



UNIVERSIDAD DE MÁLAGA



GRADO EN INGENIERÍA INFORMÁTICA

HERRAMIENTA PARA EL ANÁLISIS TEMÁTICO DE
OPINIONES EN TWITTER

A TOOL FOR TOPIC-BASED OPINION ANALYSIS VIA
TWITTER

Realizado por
FRANCISCO JESÚS GÁMEZ RUIZ

Tutorizado por
EDUARDO GUZMÁN DE LOS RISCOS

Departamento
DEPARTAMENTO DE LENGUAJES Y CIENCIAS DE LA
COMPUTACIÓN

UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO 2020

Resumen

Hoy en día es muy común usar las redes sociales para analizar todo lo que sucede a nuestro alrededor. Esto supone, que a lo largo del día se están publicando una cantidad relevante de información. Uno de los aspectos más importante que se puede encontrar en la información que se recoge en las redes, son las opiniones de los usuarios.

El objetivo de la herramienta que se va a desarrollar en este Trabajo de Fin de Grado tiene en cuenta todo lo que se ha comentado anteriormente. Esta se conectará con los servicios de Twitter, una red social que cuenta con más de 145 millones de usuarios, para recopilar la información que se está publicando sobre opiniones. Estas opiniones de los usuarios se centran en diferentes temáticas. Una vez recopilada la información, se realizará un estudio para conocer la polaridad del sentimiento que tienen las opiniones que se han recogido, teniendo en cuenta la temática seleccionada. La herramienta mostrará también un análisis global y visual sobre la temática en cuestión, para conocer si la opinión en general es buena o mala y así ayudar a los usuarios a la toma de decisiones sobre un tema concreto.

Palabras clave:

Twitter, análisis de sentimientos, Python, Django.

Abstract

Nowadays it is very common to use a social network to analyze everything that happens around us. This means that a significant amount of information is being published throughout the day. One of the most important aspects that can be found in the information that is collected on the networks is the opinions of the users.

The objective of the tool to be developed in this Final Year Dissertation takes into account everything that has been mentioned previously. This will connect with the services of Twitter, a social network that has more than 145 million users, to collect the information that is being published about opinions. These user opinions focus on different themes. Once the information has been collected, a study will be carried out to find out the polarity of the opinions that have been collected, taking into account the selected theme. The tool will also show a global and visual analysis on the subject in question, to know if the opinion in general is good or bad and thus help users to make a decision on a specific topic.

Keywords:

Twitter, sentiment analysis, Python, Django.

Índice

Resumen	1
Abstract	1
Índice	1
Introducción	1
1.1. Presentación	1
1.2. Motivación	2
1.3. Objetivos	2
1.4. Estructura de la memoria	2
Estado del arte	5
2.1. Contextualización	5
2.2. Estudio del mercado.....	7
2.3. Objetivos de la aplicación.....	9
2.4. Metodología	10
2.5. Hitos del proyecto.....	13
Análisis y Especificación de Requisitos	15
3.1. Requisitos del sistema.....	15
3.1.1. Requisitos funcionales.....	15
3.1.2. Requisitos no funcionales.....	17
3.1.3. Requisitos de información.....	17
3.2. Diagramas.....	18
3.2.1. Tema	18
3.2.2. Categoría	19
3.2.3. Tweets	20
3.2.4. Estadísticas	21
Herramientas para el desarrollo	23
4.1. Colección de herramientas	23
4.2. Concepto de API.....	23
4.3. API Twitter.....	25
4.4. Herramientas de análisis de sentimientos.....	26
4.5. Herramientas para el sistema	30
4.6. Utilidades para el desarrollo.....	32
Diseño de la herramienta	35
5.1. Arquitectura de la herramienta	35
5.2. Diseño de la web.....	36
5.3. Diseño de la base de datos	42
5.4. Realización de las estadísticas	44
Implementación	45
6.1. Inicio de la implementación.....	45
6.2. Desarrollo del script	47

6.3. Base de datos	52
6.4. Diseño Web	52
6.4.1. Front-End	52
6.4.2. Back-End	53
Test y pruebas	57
7.1. Test	57
7.2. Pruebas	59
Conclusiones.....	65
8.1. Problemas encontrados	65
8.2. Líneas futuras	65
8.3. Conclusiones	67
Referencias.....	69
Manual de Instalación.....	71
Introducción.....	71
Instalación de MongoDB.....	71
Instalación Django.....	72
Ejecución de la herramienta.	72
Guía de uso.....	73
Introducción.....	73
Preparación del entorno.....	73
Herramienta.....	75

1

Introducción

1.1. Presentación

Las redes sociales tienen un fuerte auge dentro de la sociedad, cada día más gente hace uso de ellas en su vida cotidiana. También son utilizadas para la creación de contenido, anunciarse como empresa, mostrar la opinión sobre temas diversos que ocurren alrededor. Unos de los datos que hacen pensar que es necesario el estudio de las redes sociales es tener en cuenta toda la cantidad de usuarios que están registrados en alguna red social y están activos en ella. Si cada uno de los usuarios activos publicase alguna opinión a lo largo del día, podemos concluir que, hay millones de datos en la red.

Twitter es una red social en la que cualquier usuario puede publicar una cantidad de texto, imágenes, vídeos. Uno de los aspectos claves de esta aplicación, es el uso que se suele dar, opinar sobre cosas que están sucediendo sin ningún tipo de tapujo. Se utiliza esta aplicación cuando te han sucedido altercados, como por ejemplo, si se ha sido tratado de una manera incorrecta por alguna empresa o entidad y se quiere dar a conocer el trato a los clientes, o si en cualquier caso, ha sucedido al contrario y se quiere poner una opinión positiva para que más gente haga uso de ella. Dentro de las opiniones que se pueden encontrar en esta red social, se pueden dividir en temáticas. Para conocer a qué temática se está refiriendo la opinión que contiene el mensaje, se suele hacer uso de unas palabras conocidas como hashtag o simplemente mencionando aquello a lo que se quiere referenciar.

A la hora de elegir algo, se suele usar las opiniones de personas que ya han experimentado y conocen los beneficios o inconvenientes que pueden aparecer. Para llevar a cabo una recogida de opiniones de gente conocida que se tenga el contacto para poder hacer cualquier pregunta sobre el tema en cuestión, puede llegar a ser bastante duradero y si no se obtiene una información relevante, puede que llegue a ser una pérdida de tiempo. Es más interesante realizar una recopilación de datos sobre la aplicación de Twitter para obtener las opiniones de millones de personas sin tener que realizar ningún esfuerzo y obtener un resultado claro y conciso.

1.2. Motivación

Uno de los aspectos más importantes para la sociedad es la capacidad de poder mostrar las opiniones libremente. Actualmente, existe una facilidad para informar sobre los aspectos que nos rodean, se tiene la capacidad para añadir comentarios cuando se realiza una compra por internet para que los demás usuarios conozcan las ventajas y desventajas que existen sobre el producto, también se encuentran páginas para recoger la información relevante a las opiniones, pero una de las formas que más se utiliza es mostrar las opiniones a través de las redes sociales. La dificultad que se encuentra es el análisis de esas opiniones, debido a que se necesita observar uno a uno las opiniones para conocer el criterio global. Una solución a este problema es la realización de un análisis a través de computadoras. Esta es la dirección que tomará este trabajo que además es una opción en auge en el mundo empresarial, con el objetivo de obtener información para ayudar en la toma de decisiones.

1.3. Objetivos

Uno de los aspectos fundamentales a la hora de realizar cambios es el conocimiento de las opiniones de personas que hayan tenido la experiencia con los diversos temas que nos rodean. En el pasado, el conocimiento de estas opiniones estaban basados en entrevistas a la gente de a pie a través de llamadas telefónicas, cartas o correos electrónicos. Uno de los pasos fundamentales que se han dado con el tiempo es la realización de estas cuestiones mediante computadores y su avance ha sido tan extenso que uno de los principales problemas es recabar y procesar toda esa información. La solución a esto es el desarrollo del estudio de análisis de sentimientos a través de computadoras para automatizar los procesos.

A lo largo de este documento, se realizará un estudio de los procesos para realizar el análisis de sentimientos con el objetivo del desarrollo de una herramienta que permita la recopilación de datos extraídos a través de la red social Twitter, en los cuales, se identifica la polaridad positiva, negativa o neutra de las opiniones de los usuarios.

1.4. Estructura de la memoria

Para cumplir con los objetivos se ha desarrollado una estructura secuencial según se han ido realizando los hitos del proyecto. Se han elaborado capítulos diferenciados entre sí en función de su contenido y su relación con las tareas identificadas.

- **Capítulo 1: Introducción**

A lo largo de este capítulo se hará una pequeña introducción del sistema a desarrollar, poniendo en situación al lector sobre lo que se va a ver en el documento. Se incluye un apartado en el cual se analiza la motivación que ha llevado a la realización de esta herramienta. Es importante también explicar el objetivo principal de este documento, y por tanto, se incluye en este capítulo junto con la estructura de la memoria.

- **Capítulo 2: Estado del arte**

Se lleva a cabo un análisis contextual actual sobre el análisis de sentimientos. Acto seguido, se mostrará un estudio sobre las aplicaciones existentes en el mundo empresarial para conocer el mercado. A continuación, se abordará el objetivo que se pretende que cumpla la aplicación y metodología e hitos de trabajo que se cumplirán.

- **Capítulo 3: Análisis y Especificación de Requisitos**

En este capítulo, se observarán las especificaciones que debe de cumplir el sistema a desarrollar, para esto, se introducirán:

- Requisitos: Objetivos que la aplicación debe cumplir.
- Diagramas: Desarrollo sobre cómo se implementarán los requisitos.

- **Capítulo 4: Herramientas para el desarrollo**

Este capítulo contendrá la explicación de qué es una API Rest, el estudio de las API para encontrar la más conveniente para nuestra herramienta, el desarrollo explicativo de la función de la API de Twitter que nos permitirá la recopilación de los tuits. También se tendrán en cuenta todas las herramientas que se utilizarán para la elaboración de la aplicación, así como, los lenguajes y los entornos de desarrollo utilizados.

- **Capítulo 5: Diseño de la herramienta**

En este capítulo se describe la arquitectura que seguirá el sistema a desarrollar y por qué se ha elegido. A parte de la estructura, también se pueden encontrar las maquetas de la aplicación, es decir, el diseño que seguirá la aplicación, así como, la web a desarrollar, la base de datos y demás componentes necesarios para su funcionamiento.

- **Capítulo 6: Implementación**

Este capítulo es donde se describe el proceso de desarrollo de la aplicación, que se estructura en los siguientes apartados:

- Desarrollo de script: Se describe la implementación de los script, los cuales, se centran en la conexión con las herramientas necesarias de Twitter, la recopilación de los tuits, limpieza de los datos y el análisis de los mismos.
- Base de datos: Creación de la base de datos, conexión con la aplicación y puesta en marcha para el almacenamiento de los datos recopilados.
- Desarrollo web: En este apartado se explica el proceso de elaboración de la aplicación web. Para ello, se abordan las dos partes del sistema, una donde se explica el front end de la web, las características visuales, y la siguiente, sobre el desarrollo del back end, donde se encuentra la lógica de la aplicación, conectando los servicios realizados anteriormente con la web.

- **Capítulo 7: Test y pruebas**

Este capítulo es fundamental, puesto que describe el proceso de comprobar que todo funcione correctamente. Primero, se comprueba que la aplicación recopila perfectamente los tuits que se le requieran. Segundo, se observa que se desarrolla sin ningún tipo de problema el análisis de sentimientos de los datos de la primera fase. A continuación, se observa el funcionamiento de la web con las estadísticas que se proponen.

- **Capítulo 8: Conclusiones**

Resumen de las conclusiones sobre el desarrollo de la aplicación, el uso del lenguaje Python para la solución obtenida en los script, todos los conocimientos adquiridos sobre MongoDB, el uso de las API, y la utilización de Django para la elaboración de la aplicación web. Como último apartado, se comenta sobre aspectos futuros de la aplicación y posibles mejoras de la misma.

2

Estado del arte

2.1. Contextualización

El análisis de sentimientos, o minería de opinión, es el uso del procesamiento del lenguaje natural o métodos de lingüística computacional que ayudan a conocer, extraer e identificar información subjetiva del contenido existente en los textos. Dicho de otra manera, es el área de estudio que analiza las opiniones de las personas que reflejan sobre entidades que les rodean, estas pueden ser, productos, lugares, empresas y todo tipo de temas. Unos de los aspectos fundamentales para la utilización del análisis de sentimientos es el uso en redes sociales, ya que permiten hacer una idea de la opinión pública que hay en general sobre un determinado tema.

Para realizar un estudio de los sentimientos, hay que conocer dos atributos claves que deben aparecer en la información recabada: el objetivo, que hace alusión sobre el tema que se está abordando y, como ya se ha mencionado, el sentimiento, que es el que caracteriza la posición en la que se encuentra la opinión.

Con el análisis de los sentimientos, se persigue lograr entender la intención que un sujeto pone sobre cierta frase. Acto seguido, se quiere conocer la valoración que tiene la frase, para ello hay que tener en cuenta la polaridad que contiene, con la cual, se clasificará el sentido del mensaje pudiendo ser positivo, negativo o neutro.

La primera idea que se viene a la cabeza al pensar sobre el análisis de opiniones y sentimientos es que si un texto contiene adjetivos positivos, la opinión será positiva, y si un texto contiene adjetivos negativos, evidentemente, la opinión que se está reflejando en la oración es negativa. No se está muy lejos cuando se piensa eso, no obstante, aún hay que tener en cuenta más datos.

Se encuentran varios enfoques para la minería de opinión. Se puede hablar de los enfoques semánticos, estos son caracterizados por diccionarios de términos con orientación de opinión o polaridad. Para el desarrollo de este enfoque, se separan las

palabras que contiene el texto que se está analizando, con el objetivo de eliminar palabras que no suman ni restan a la polaridad del sentimiento, dejando así, las palabras clave que están contenidas en el diccionario. Suelen ir acompañados de identificación de adverbios que hacen una mayor entonación al sentimiento que el usuario quiere reflejar. El siguiente aspecto a conocer es el aprendizaje computacional que consiste en entrenar un clasificador usando un algoritmo de aprendizaje. A este se le hace llegar textos, que se conocen su polaridad, para ir entrenando y así conozca y sepa reconocer el sentimiento.

Otro de los aspectos fundamentales a la hora de realizar un análisis es conocer la zona de estudio. Como se ha comentado, los análisis se realizan sobre textos que se encuentran en la red, con lo cual, hay expresiones coloquiales. Esta característica demanda lo que se conoce como procesamiento del lenguaje natural, este es el encargado de procesar los textos que están escritos coloquialmente y los traduce en un lenguaje que la máquina sea capaz de entender.

El motor, encargado de realizar este estudio, recopila una gran cantidad de datos para la obtención más precisa y así realizar un mejor desempeño en los algoritmos a la hora de obtener la información que procesa mediante el proceso del lenguaje natural.

La minería de opinión se enfoca dependiendo de las diferentes perspectiva según el análisis que se requiera. Según esto podemos ver los siguientes niveles:

- **Categorización del contenido:** realiza un breve resumen del contenido del texto según la lingüística utilizada, incluyendo alertas de contenido y detección de duplicación. Captura los significados y los temas en los que se abarca el texto para un análisis avanzado y predicción. Este tipo de análisis es efectivo en textos o documentos los cuales hablan siempre sobre un tema en concreto y no sobre los que realizan varias opiniones a modo de comparativa. Se pueden observar en los estudios de Pang, Lee y Vaithyanathan en 2002 y Turney en 2002. [1]
- **Singularidad del contenido:** este nivel es el encargado de realizar el análisis expresado en cada frase. Obtiene el texto a analizar y separa el contenido en frases, las cuales, son evaluadas por separado. Observando el estudio de Wiebe, Bruce y O'Hara en 1999 [2], donde esta caracterización viene referenciada en la clasificación subjetiva.

Teniendo en cuenta todo lo anterior, también hay que diferenciar algunos aspectos que usamos los humanos a la hora de mostrar nuestras emociones y sentimientos, estos pueden ser bastante dificultosos a realizar un análisis de sentimientos. Hablamos sobre:

- **Contextualización:** Hay algunas expresiones en el lenguaje coloquial que dependiendo del contexto, de las personas con las que nos estemos comunicando o sobre la entidad a la cual nos referimos, pueden significar algo positivo o negativo. Uno de los ejemplos más claros que se puede ver es cuando nos referimos a nuestras parejas con el término coloquial *gordo o gorda*; si lo analizamos de forma descontextualizada, la palabra puede ser tanto negativa

como positiva, ya que hace referencia a una característica, posiblemente, ofensiva sobre una persona, aunque usualmente se utiliza como algo cariñoso.

- **Ironía:** Una ironía es una frase usada en el mundo coloquial para referirse a todo lo contrario que se está expresando. Es algo muy usado a la hora de realizar una queja. También se suele dar a conocer como sarcasmo o burla, debido a que es una forma de mofa. Se puede ver, por ejemplo, cuando nos estamos quejando de la conexión de internet, en este contexto, se suele decir cosas tales que, *Que bien me va el internet, esta compañía es la mejor*. De primera hora parecen frases totalmente positivas, pero son irónicas, con lo cual, tienen una polaridad totalmente contraria.

La posible solución para este problema es que a la hora de obtener esa información, el sujeto que haya escrito la opinión, deje dicho que se trata de un sarcasmo. Otro de las posibles soluciones es realizar un diccionario donde aparezcan diferentes opiniones similares y que se conozcan que son ironías y, así, poder observar en las recopiladas si son del mismo ámbito. [3]

2.2. Estudio del mercado

Una vez conocidos los aspectos que rodean al análisis de sentimientos, el siguiente paso es hacer un estudio de mercado para dar a conocer las herramientas ya existentes que se utilizan en el mundo empresarial. Para este estudio, se tomarán aplicaciones cuyo funcionamiento trate sobre análisis de sentimientos en redes sociales, debido a que de esto tratará el trabajo.

- TalkWalker

Es una herramienta que permite la monitorización de las redes sociales. Dentro de muchas de sus funciones, la que se ha de destacar es su función para el análisis de sentimientos. Esta permite un análisis en más de un idioma y genera a su vez estadísticas para conocer la opinión que hay de tu empresa, producto o marca. [4]

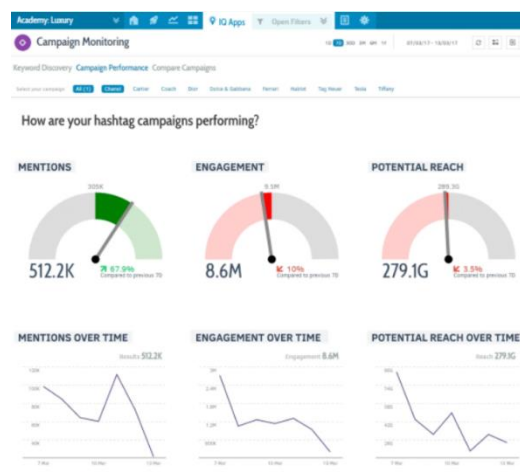


Imagen 1 Muestra de la aplicación TalkWalker

- Brand24

Esta aplicación de pago permite conocer las opiniones buscando palabras claves en las redes sociales, como, por ejemplo, el nombre de la empresa, dirección. Permite el análisis de las redes sociales, métricas de compromiso observando el seguimiento que se obtiene y permite el análisis de sentimientos. A través de todo esto, genera estadísticas para que se observe con detenimiento la opinión general de la empresa. [5]

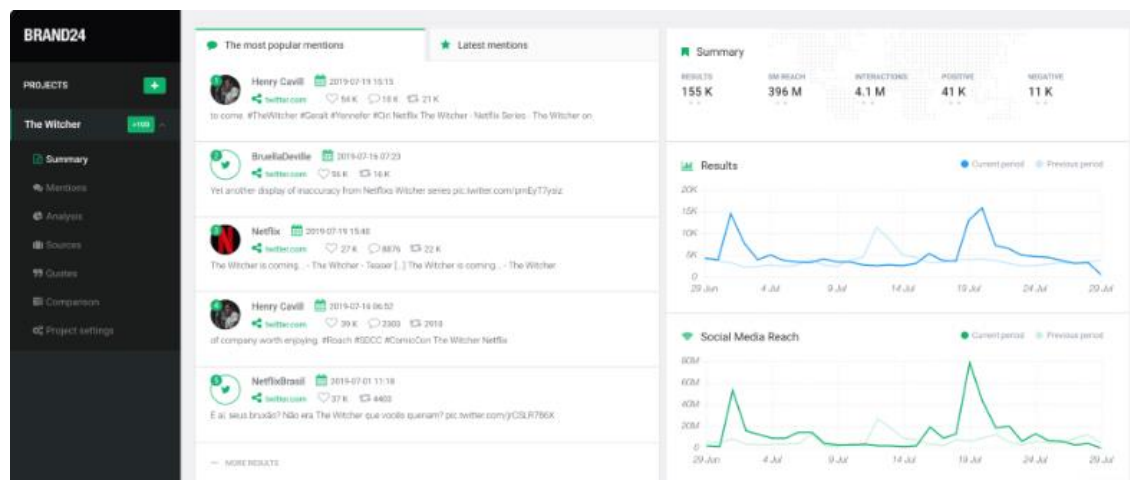


Imagen 2 Muestra de la aplicación Brand24

- Keyhole

Esta aplicación realiza una búsqueda por etiquetas o sobre nombre de cuentas de redes sociales. Además, también es utilizada para realizar un seguimiento a las menciones sobre los perfiles que se introducen. Permite hacer una búsqueda de datos históricos que hay sobre la red social, el número de seguidores, comentarios, participación y personas influyentes. Realiza informes con estadísticas a tiempo real, datos históricos y también aplica análisis de sentimientos sobre las opiniones que encuentra en la red. [6]



Imagen 3 Muestra de aplicación Keyhole

- Rapidminer

Es una aplicación que permite la conexión con la red social Twitter para realizar una recopilación de tuits. Una vez recopilados los tuits, la aplicación permite procesarlos para añadir un análisis de sentimientos. [7]

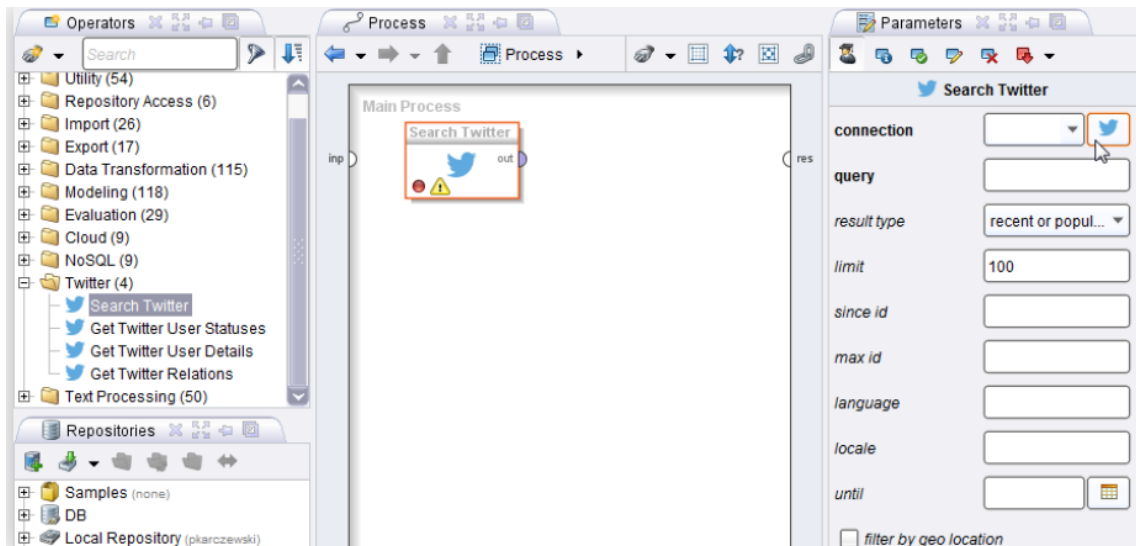


Imagen 4 Muestra de la aplicación RapidMiner

2.3. Objetivos de la aplicación

La aplicación que se propone para este trabajo es la elaboración de una aplicación web que permita la recopilación de tuits basado en temáticas y realice un análisis de sentimientos para que sea capaz de crear estadísticas y así utilizarlo como método para la ayuda en la toma de decisiones. Para ello, la aplicación debe cumplir los siguientes objetivos:

- **Elección de temática:** Cuando se inicie la aplicación, aparecerá una pestaña, en la cual, el usuario pueda seleccionar el tema sobre el que desea hacer el análisis.
- **Elección de categorías:** Una vez elegida la temática, aparecerá la opción de categorías, sobre las cuales, quiere buscar la información.
- **Recolección de los datos:** Los datos, que se recolectan a través de la aplicación de Twitter, van relacionados con las características que el usuario ha introducido anteriormente, es decir, si el usuario ha elegido un tema y una categoría, la aplicación es la encargada de buscar los tuits que se encuentran en Twitter que tengan que ver sobre ese ámbito. Habrá dos opciones, una que permite la recopilación de tuits que hay históricos en la red social y la otra opción es sobre los tuits que se están publicando desde que se ejecuta la búsqueda.
- **Análisis de los datos recolectados:** Una vez recopilados los datos necesarios y pedidos por el usuario, se realizará un análisis de sentimientos para clasificar los diferentes tuits según su polaridad y así obtener resultados sobre las opiniones globales de los usuarios de la red social. Gracias a las opciones de búsqueda, permite hacer comparativas sobre la evolución que tienen los diferentes temas a buscar, así comparar con los cambios realizados y observar si ha habido mejoras en comparación al pasado. Una vez realizado el análisis, la aplicación ha de ser capaz de identificar las diferentes categorías que el usuario ha

seleccionado y así poder observarlas en una misma vista para la ayuda en la toma de decisiones.

- **Observar las estadísticas:** Otro de los aspectos que debe tener la aplicación, es observar los datos y el estudio que se ha realizado. Para ello, debe haber un apartado donde se encuentren todas las estadísticas elaboradas sobre las diferentes categorías. Para una mayor visualización, se debe mostrar, mediante gráficas, la comparativa de las opiniones que se han encontrado en la red social.

2.4. Metodología

Para el desarrollo de una aplicación software, una de las cosas más importantes es definir qué metodología se va a seguir. La metodología es un conjunto de procedimientos, técnicas y soporte documental que ayuda a los desarrolladores a, en este caso, realizar un nuevo software. Para el desarrollo de sistemas software se pueden encontrar diferentes metodologías, de las cuales, se identifican las siguientes:

- **Modelo de cascada**

Este es un método que se considera como el tradicional que explica el proceso de desarrollo de los sistemas en ingeniería del software, por lo que actualmente es visto como anticuado. Es un proceso que se tiene en cuenta cuando se conoce perfectamente las metas y los requisitos. Está dividido por etapas, cuando una etapa termina empieza la siguiente y no puede volverse atrás, con lo que, no es un buen método para el desarrollo del TFG.

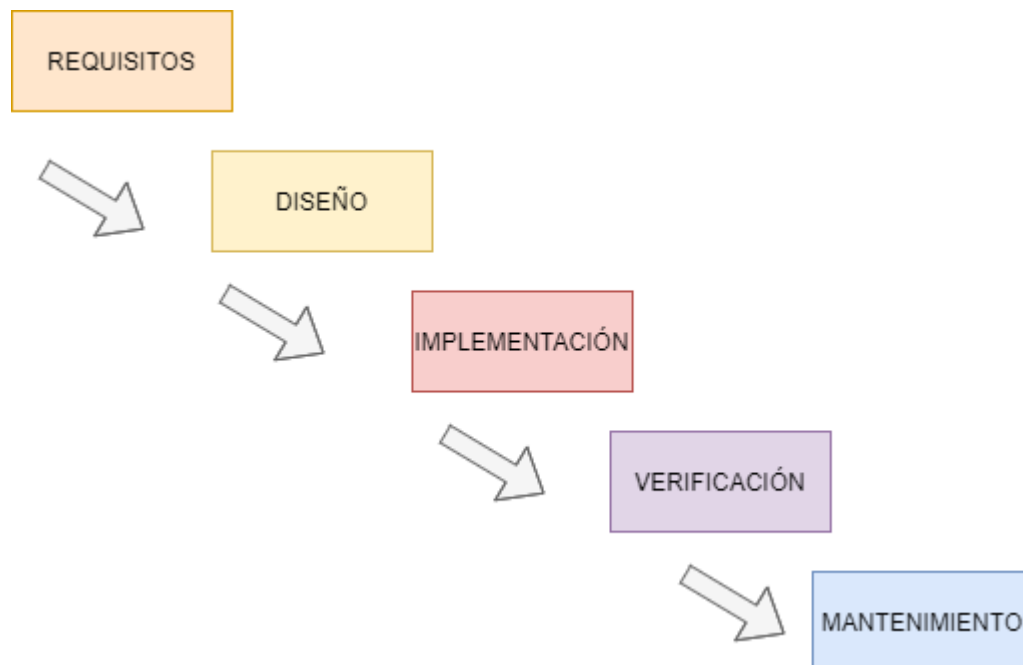


Imagen 5 Fases del modelo cascada

- **Modelo de espiral**

Este método da un paso evolutivo del modelo anterior, añade consigo la funcionalidad de incrementos e iteraciones que funcionan correctamente cuando se trata de un desarrollo de un sistema software. En cada iteración o bucle representa un conjunto de actividades, como las que pueden llegar a ser prototipos rápidos, mayor paralelismo y concurrencia de actividades. Es un buen método para tener en cuenta para el desarrollo.

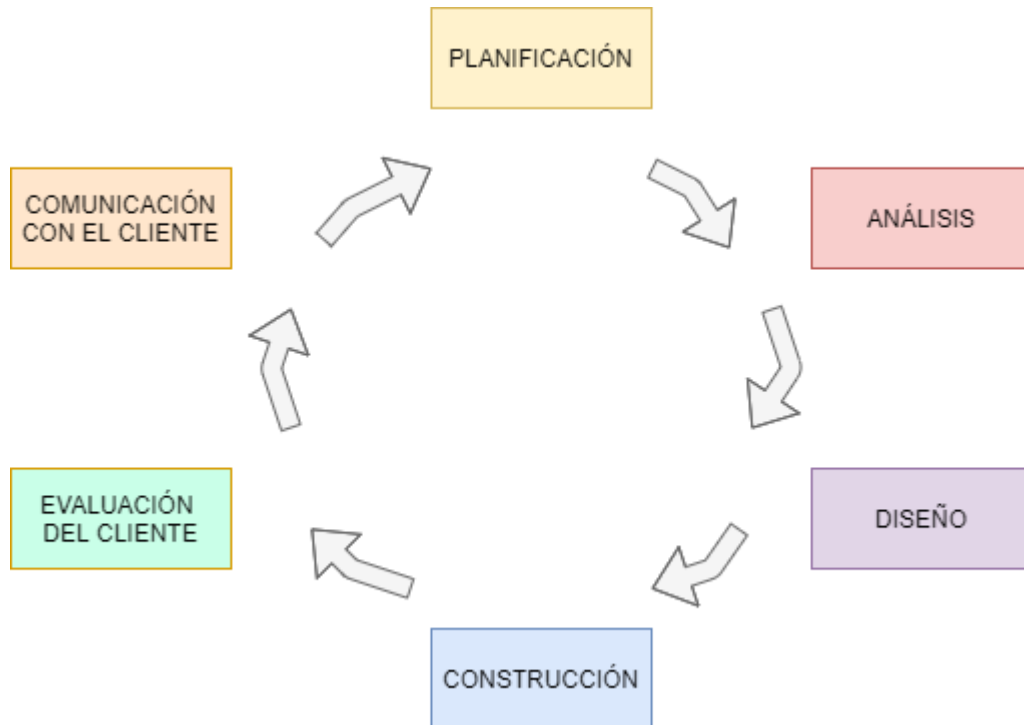


Imagen 6 Fases del modelo de espiral

- **Metodología de prototipo**

Es un procedimiento de desarrollo que permite que los desarrolladores creen un primer prototipo de la solución de la herramienta a realizar para, así, que la parte interesada pueda observar las funcionalidades de la misma y poder hacer los cambios que necesite antes de la creación final. Una de las ventajas que se obtienen con esta metodología que da una idea bastante clara de lo que pretende el proyecto ya que desde muy principio de su desarrollo se pueden observar las funcionalidades que esta tendrá. Este método no es recomendable para el desarrollo de este proyecto.

- **Programación extrema**

Es una metodología creada por Kent Beck, la cual, viene desarrollada en su libro *Extreme Programming Explained: Embrace Change* [8]. En él se explica que fue desarrollado para abarcar las necesidades específicas sobre proyectos pequeños dirigidos por equipos pequeños. Los requisitos del sistemas son ambiguos y cambiantes, con lo que, tienen un alto riesgo. Uno de los aspectos fundamentales de la programación extrema, es que un equipo se encarga de la programación mientras que otro equipo se mantiene cerca de los programadores para posibles correcciones, lo esencial es que los equipos se conozcan con antelación para el perfecto flujo del desarrollo. Este no es un método

aconsejable para el Trabajo de Fin de Grado, ya que, se conoce perfectamente los requisitos que debe de cumplir.

- **Scrum**

Scrum fue identificada por Nonaka y Takeuchi y se puede encontrar en su libro *The New Product Development Game* [9]. Scrum es una colección de componentes y partes obligatorias que están definidas. Está basado en un proceso empírico e iterativo con respecto a las personas y se basa en la autoorganización de los equipos para la resolución de los problemas inspeccionando y adaptándose continuamente. Aparecen eventos para fomentar la comunicación entre los equipos, pero haciéndolo más fácil y rápido que las reuniones para que sea más dinámico. Este aspecto lo destaca sobre los demás, ya que, permite un contacto permanente con el cliente y esto hace que todas las partes conozcan la evolución del proyecto. Se ha elegido esta metodología, ya que, es la más indicada para este tipo de proyectos, debido a que permite un contacto permanente con las partes para así conocer con exactitud los objetivos, la evolución y el desarrollo del proyecto.

Durante la visualización de las metodologías que se ha desarrollado anteriormente, se ha seleccionado como la más adecuada la Scrum. Uno de los aspectos fundamentales para esta idea o metodología software son la identificación clara de las partes. Para ello, se conocerán los procesos y las partes que menciona Scrum y se identificarán.

- **Procesos**

La forma de actuación se realiza de forma iterativa. Cada repetición se denomina *sprint*, normalmente, tiene una duración entre 2 a 4 semanas, para este proyecto, se elige, como mejor opción, una duración de 2 semanas para los sprint para tener un mayor contacto con los entregables debido a que el desarrollo del sistema hay que realizarlo en un periodo corto.

- **Product Backlog:** Es el conjunto de los requisitos que se identifican en el sistema a los que se le identifica la dificultad obteniendo el tiempo que llevarán consigo las diferentes tareas. Este conjunto se definirá cuando se visualicen los diferentes hitos del proyecto y obteniendo el tiempo que se requiere para la realización de cada uno de ellos.
- **Sprint planning:** Este proceso es el que ordena las tareas que se han definido en el Product Backlog y se elabora una decisión del número de tareas que se pueden realizar para cada sprint. Se verá identificado en los hitos del proyecto.
- **Sprint:** Son las iteraciones que tendrán el proyecto, tienen una duración definida para todo el proyecto y estarán convocados todos los componentes del equipo que trabaja en el sistema. Al final de cada sprint, se deberán de entregar diferentes versiones de la aplicación.
- **Sprint Backlog:** Están definidas las diferentes tareas que se tiene que realizar en los sprint.
- **Daily sprint meeting:** Reunión diaria para conocer el proceso de las tareas de la semana que se han ido realizando a lo largo del día. Este proceso se mantendrá con una conversación diaria con el tutor para conocer la evolución del proyecto.

- **Demostración:** Se muestra la evolución que está llevando el proyecto. Esta etapa aparecerá en el final de cada sprint. Se observará en las reuniones con el tutor todo el proceso que se ha realizado durante las semanas de sprint.

- **Roles**

Otro de los aspectos importantes de esta metodología son los roles, estos son identificados y caracterizados, con lo que, se observarán los diferentes perfiles que hay dentro de Scrum y se identificarán, de forma aproximada, las partes que se componen para el desarrollo de este Trabajo de Fin de Grado. [10] [11]

- **Scrum master:** Esta es la persona encargada de liderar el equipo, que se cumplan las reglas establecidas y los procesos de la metodología. Este perfil está asignado al alumno que está realizando este proyecto, debido a que se tiene que encargar de que todo vaya correctamente y se estén cumpliendo las estrategias.
- **Product owner:** Representante del cliente, es el encargado de transmitir los requisitos del sistema, las reacciones del cliente, aclarar las dudas del lenguaje utilizado, fijar fechas de entrega, valida las entregas y los sprint. Este perfil estará identificado por el tutor, ya que será el encargado de revisar el trabajo realizado por el alumno y validar todo el proceso realizado.
- **Equipo:** Equipo de personas que trabajan en los procesos software y cumplen con los hitos que se proponen. El alumno es el que forma el equipo, porque es el encargado de realizar el trabajo y la solución software.
- **Stakeholder:** Es la parte interesada o cliente, es el encargado de proporcionar los requisitos y validar los resultados. Este trabajo está siendo revisado por el tutor, debido a que conoce los objetivos y tiene una mayor visión sobre los objetivos del sistema, con lo cual, este rol estará identificado por el tutor.

2.5. Hitos del proyecto

En este apartado se encuentran las fases del proyecto que se sigue para la obtención de la solución de la herramienta que se ha planteado. Para ello, se muestra las fases que se van a llevar a cabo junto con una pequeña descripción para contextualizar cada una de ellas y así, poder relacionar las horas que deben de cumplir cada una de ellas.

1. Obtención de requisitos

Esta fase consistirá en la obtención de los requisitos del sistema. Mediante la observación de los objetivos que se han planteado, se realiza la búsqueda de los requisitos que ha de tener el sistema para llegar a la mejor solución para la herramienta.

2. Elaboración de diagramas

Una vez analizado los requisitos necesarios para la aplicación, se desarrollarán los diagramas que debe de seguir la aplicación y así comprender el uso y el funcionamiento que debe de seguir la herramienta.

3. Desarrollo de Scripts

Esta fase del proyecto se dividirá en tres:

3.1 Extracción de los datos

En esta fase se recopilan los tuits que son necesarios para la realización del análisis de sentimientos sobre los temas que se seleccionarán. Para ello, se debe de realizar una conexión con la aplicación de Twitter que permitirá el acceso a los tuits publicados por los usuarios.

3.2 Conexión con la base de datos

En esta fase se realizará el almacenado de los datos que se han recopilado en la fase anterior en la base de datos.

3.3 Limpieza de los datos

Como toda información que se obtiene a través de los tuits, se conoce que están escritos en un lenguaje natural debido a que usan un lenguaje coloquial, se conoce que hay datos y palabras que no se pueden tratar. En esta fase del proyecto es la encargada de limpiar los datos para que estén listos para analizar.

3.4 Procesamiento de texto

Una vez realizada la limpieza de los datos que se han recopilado, hay que tratar la estructura de esta información y obtener lo relevante a la hora de analizarlos.

3.5 Análisis de datos

En esta fase se realizará un análisis de los datos obtenidos en las fases anteriores. Gracias al proceso de limpieza, la aplicación será capaz de identificar la polaridad de las opiniones de los usuarios y así, cumplir con el análisis requerido.

4. Desarrollo web

Se realizará el desarrollo de la web que nos permita la visualización de los datos para así conocer los diferentes aspectos a tratar y la opinión de toda persona usuaria de Twitter.

5. Documentación

Elaboración de la memoria del proyecto, la presentación y la defensa del Trabajo de Fin de Grado.

3

Análisis y Especificación de Requisitos

3.1. Requisitos del sistema

Para el desarrollo de cualquier aplicación que se vaya a realizar, es necesario tener en cuenta cuáles son los objetivos que se quieren alcanzar con la misma. Para ello, hay que realizar un estudio de los requisitos para tener en cuenta a la hora llegar a la mejor solución para el problema.

3.1.1. Requisitos funcionales

Los requisitos funcionales son aquellos en los cuales se declaran los servicios que cumplirá el sistema o la forma en la que actuará la aplicación. Dentro de los requisitos se reflejan las acciones que podrá realizar el usuario cuando esté usando la aplicación o las acciones que hace la aplicación con otros sistemas. Véase los diferentes requisitos funcionales que se tendrán para la solución:

- RF1: Elegir temática:

Este requisito es uno de los fundamentales para la aplicación. En este caso, se encargará de seleccionar la temática para la cual se quiera hacer el análisis de sentimientos. Para ello, dentro de este requisito aparecerá el siguiente subrequisito:

- RF1.1: CRU Tema: Permitirá realizar las siguientes funciones sobre el tema:
Crear el tema el cual se querrá observar el análisis, leer el tema que ya se ha registrado y actualizar el tema a tratar.

- RF2: Elegir categoría:

Una vez elegida la temática, sobre la cual se realizará el estudio, el siguiente objeto a destacar es la categoría a la que pertenece dentro de la temática. Esta destaca debido a que se centrará más el dominio del análisis. Dentro de este requisito se obtiene el siguiente:

- RF2.1: CRU Categoría: Este requisito permite realizar las funciones de: Crear la categoría que se quiere evaluar, leer la categoría que está en el sistema y actualizarla.

- RF3: Recopilación de los datos:

Este es uno de los requisitos más importantes de la aplicación, debido a que el análisis se realizará sobre los datos que se irán recolectando gracias a este. Para ello hay que tener en cuenta el siguiente subrequisito:

- RF3.1: CR Tweets: Permitirá que una vez que se esté realizando la recogida de datos, se pueda realizar las siguientes funciones sobre él: crear el tweet en la base de datos con los datos recogidos y leer los tweets que se encuentran dentro de la base de datos.

- RF4: Limpieza de los datos:

Este requisito es el encargado de realizar una limpieza de los datos, debido a que a la hora de recopilar la información se obtienen datos innecesarios para la aplicación. Se realizará un proceso de estudio para conocer la información necesaria a almacenar y una vez eliminada la que sea prescindible, se almacenará aquella interesante para analizar en la base de datos gracias al objeto tweet.

- RF5: Análisis de los datos

El requisito de análisis de datos permite observar con detenimiento las diferentes características que tienen los tuits que se han ido recopilando y almacenando en la base de datos. Para ello, se estudiará la polaridad de las opiniones que recogen los tuits.

- RF6: Visualizar estadísticas:

En este apartado, se recogen las diferentes características que se quieren mostrar una vez se hayan realizado las operaciones de extracción, transformación y carga de los tuits. Dentro de este requisito, aparecen subrequisitos necesarios para la implementación:

- RF6.1: CR Estadística positiva: Recoge los tuits positivos que se han generado con el tema en cuestión. Se tendrá en cuenta que se podrán crear y leer las estadísticas positivas para las comparativas.

- RF6.2: CR Estadística negativa: En este apartado, como aparecía en el caso de RF6.1, aparecerán los tuits negativos sobre el tema a tratar. Para realizar la estadística se creará y leerá las estadísticas negativas para las comparativas.
- RF6.3: CR Estadística neutral: Al igual que en los dos casos anteriores, se calculará una estadística basada en la neutralidad de las opiniones sobre el tema, el cual, se está tratando. Para ello, es necesario poder crear las estadísticas y hacer una lectura de las que se han creado.
- RF6.4: CR Comparativa de estadísticas: Se crea y lee una comparativa de las estadísticas que se han generado para conocer la opinión global que se tiene sobre el tema.

3.1.2. Requisitos no funcionales

Los requisitos no funcionales, como su nombre indica, son aquellos que no se refieren a la función que realizará la aplicación, sino a las propiedades que tiene que tener el sistema. Los requisitos no funcionales fundamentales para la aplicación a desarrollar son los siguientes:

- RNF1: Rendimiento:

Requisito necesario para que la aplicación soporte la carga necesaria que se le pedirá a la hora de estar recopilando los tuits y que no haya ningún problema a la hora de guardarlos o de recogerlos de la base de datos.

- RNF2: Disponibilidad:

La aplicación debe estar disponible a la hora que se le requiera y que no tenga problemas a la hora de ser ejecutada por cualquier usuario que la necesite.

- RNF3: Estabilidad:

La herramienta no puede quedarse bloqueada cuando está haciendo el ejercicio de recopilación, de limpieza o de análisis de los datos. Una vez iniciada la aplicación, esta no debe de dejar de funcionar por ningún motivo para que esté operativa en cualquier instante de tiempo.

- RNF4: Funcionalidad:

La aplicación debe de cumplir todos los requisitos funcionales que anteriormente se han comentado para que cumpla con los objetivos que se quieren alcanzar.

3.1.3. Requisitos de información

Requisitos del sistema de información que el sistema debe de realizar, gestionar o satisfacer. Para nuestra aplicación hay que tener en cuenta los siguientes requisitos de información:

- RI1: Temática:

La aplicación debe ser capaz de gestionar las diferentes temáticas, las cuales, se realizará el estudio.

- RI2: Categoría:

Como con el primer requisito de información, es necesario poder gestionar y crear las categorías que se tratarán de los temas anteriormente seleccionados.

- RI3: Datos:

Hay que tener en cuenta que la aplicación realizará análisis de datos, con lo cual, lo más importante de los requisitos de información son los datos a tratar, estos, contendrán toda la información que se recopila a través de los tuits, como, por ejemplo, el texto donde se guarda la opinión y la polaridad de los tuits. Una vez analizados se almacenan con el siguiente requisito de información.

- RI4: Tweets:

Tendrá la información del texto, el cual, se realizará el análisis para conocer su polaridad.

- RI5: Análisis:

Se tendrán en cuenta la información recogida para realizar un análisis específico para obtener las diferentes opiniones que hay del tema a tratar, y así, conocer si tiene aspectos positivos, negativos o neutros.

3.2. Diagramas

En esta sección se encuentran los diagramas de flujo de los requisitos que se han obtenido para la resolución de la aplicación. Un diagrama de flujo es una representación gráfica de un proceso o algoritmo a través de una serie de pasos estructurados para la observación del resultado como un todo. La representación de los diagramas ayuda a la hora de conocer el camino que ha de tomar el proceso. A continuación, se definen las sucesiones de los requisitos funcionales, ya que son los que definen las funciones que realiza la aplicación.

3.2.1. Tema

Se elabora las siguientes funciones que se obtienen para los temas. Se muestra el proceso de la creación de un tema. El siguiente proceso necesario es la lectura de una tema para que el usuario pueda acceder a ella. Finalmente, se podrá realizar una actualización de la misma.

➤ Crear tema

El usuario obtiene un formulario para la creación de un tema. En este tiene que introducir el nombre y las palabras clave. Una vez introducido, se envía el formulario y el sistema comprueba si existe algún error en él. Si no han aparecido errores, el sistema crea una nueva temática.

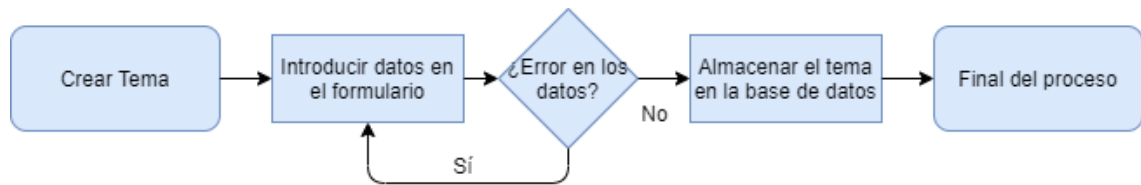


Ilustración 1. Diagrama crear tema

➤ Leer tema

Para leer el tema el sistema recoge la clave primaria del tema para identificarlo. Una vez encontrado el tema en la base de datos, se muestra la página para el usuario. Si en algún caso no existe esa clave primaria para ningún tema, el sistema devuelve al usuario un error para que introduzca una clave que se encuentre registrada en el sistema.

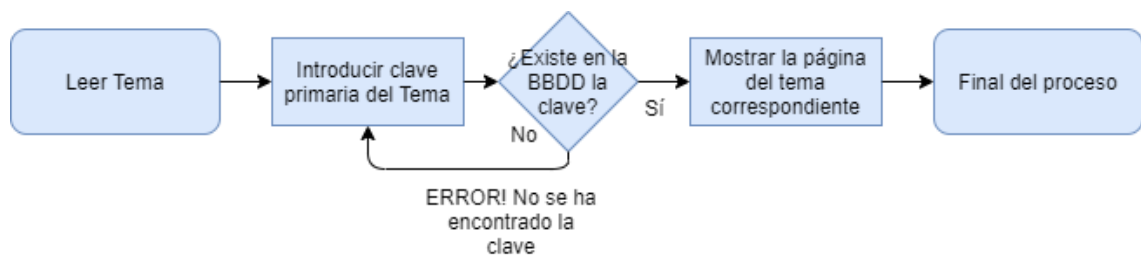


Ilustración 2. Diagrama leer tema.

➤ Actualizar tema

Para realizar la actualización del tema se introducen la clave primaria en el sistema. El sistema se encarga de buscar en la base de datos la clave introducida. Si se encuentra, aparece un formulario para editar las palabras clave del tema. Al enviar el formulario el sistema comprueba que los parámetros introducidos son correctos para actualizarlos. Si no lo son, no los actualizan y muestra de nuevo el formulario. Si no encuentra el tema en la base de datos, muestra un error debido a que no existe.

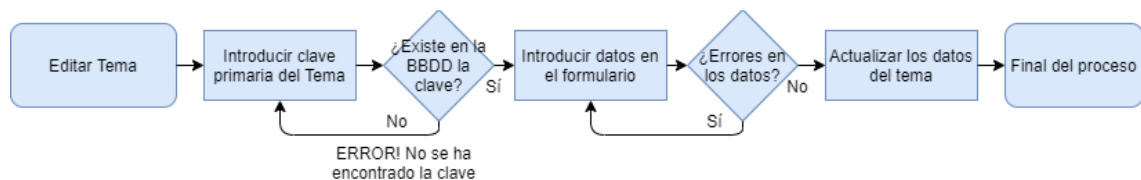


Ilustración 3. Diagrama de actualización del tema.

3.2.2. Categoría

En esta sección se presentan las diferentes funciones que se realizan sobre las categorías. Como se ha presentado en la anterior con los temas, en esta ocasión, también se podrán crear categorías, leerlas y actualizarlas. Para realizarlos, el usuario debe de seleccionar el tema. En este punto aparece el mismo diagrama realizado en la Ilustración 3, este proceso se representa mediante la acción "Buscar Tema" en lugar de la búsqueda por clave primaria.

➤ Crear categoría

Para la creación de una categoría el usuario debe de introducirse dentro de un tema e irse a la sección de actualización del tema. Aparece un formulario en el que el sistema permite la introducción de una categoría. En este caso, es necesario el nombre y las palabras clave. El sistema se encarga de validar los datos introducidos por el usuario. Si no existe ningún error en el formulario, se almacena la categoría dentro del tema seleccionado.

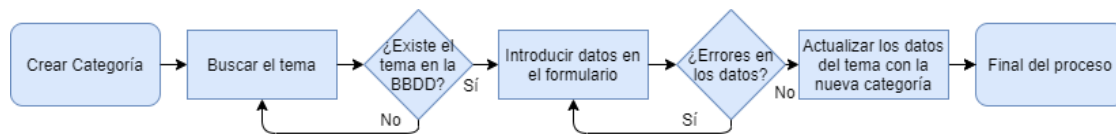


Ilustración 4. Diagrama crear categoría.

➤ Leer categoría

Para la elección de la categoría el usuario debe de introducirse en un tema. El sistema busca las categorías que tiene el tema y las muestra. El usuario selecciona la categoría que desea leer.

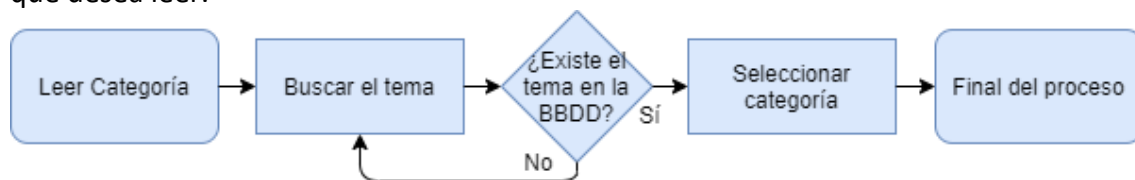


Ilustración 5. Diagrama leer categoría.

➤ Actualizar categoría

Para la actualización de una categoría el usuario debe de introducirse dentro de un tema e irse a la sección de actualización del tema. Aparece un formulario en el que el sistema permite editar los datos del tema junto a los datos de las categorías de este. El sistema se encarga de validar los datos introducidos por el usuario. Si no existe ningún error en el formulario, se actualiza la categoría dentro del tema seleccionado.

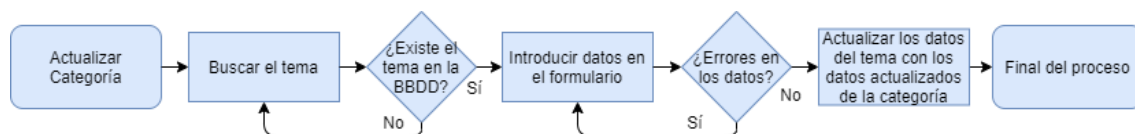


Ilustración 6. Diagrama actualizar categoría.

3.2.3. Tweets

La aplicación recopila los tuits que el usuario requiere de la red social Twitter, a su vez, esta los almacena. La herramienta genera un tweet cuando se está almacenando. La herramienta también permite la lectura de los tweets almacenados. A continuación, se observan las funciones sobre los tweets.

➤ Crear Tweet

La aplicación recoge el tuit y lo transforma de forma necesaria para que se almacene en la base de datos de la herramienta. Para ello, recibe la información a través de la

aplicación, esta se limpia y se almacena en la base de datos del sistema si no existe el tweet sobre esa búsqueda. Si se ha almacenado un tweet que tenga el mismo tipo de búsqueda y el mismo motor de análisis, no se almacena.

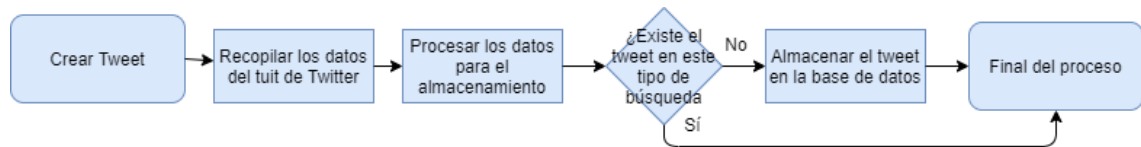


Ilustración 7. Diagrama sobre crear tweet.

➤ Leer Tweet

Para los procesos que realiza la herramienta, como la muestra de gráficas o muestra de tweets, es necesario la lectura de los tweets que se han creado anteriormente. Para ello, el sistema realiza una búsqueda en la base de datos, la cual, muestra los tweets que se han almacenado durante la recopilación. Si el sistema no encuentra datos, no puede mostrar nada y termina el proceso, en caso contrario, muestra los tweets.



Ilustración 8. Diagrama leer tweet.

3.2.4. Estadísticas

La herramienta realizará diferentes funciones referentes a las estadísticas. Las estadísticas realizadas son el porcentaje de polaridad y la evolución por días de la polaridad que existen en las categorías. La herramienta tiene que realizar la creación de ambas estadísticas sobre los temas indicados por el usuario. También se desea observar las estadísticas que se han generado anteriormente.

➤ Crear estadística

Para la creación de la estadística, es necesario recoger el análisis de sentimientos referentes a los tuits que se han ido almacenando a través de los procesos anteriores. Una vez recogido el análisis, estos se clasificarán dependiendo de la polaridad de los mismos y así realizar la estadística. El primer caso a tratar es sobre el porcentaje que existe sobre la cantidad de tweets recogidos en la búsqueda. El sistema recupera los tweets almacenados según los parámetros de búsqueda que desea el usuario. Si encuentra datos sobre esa búsqueda, se realiza un porcentaje sobre las tres polaridades que trata la herramienta.

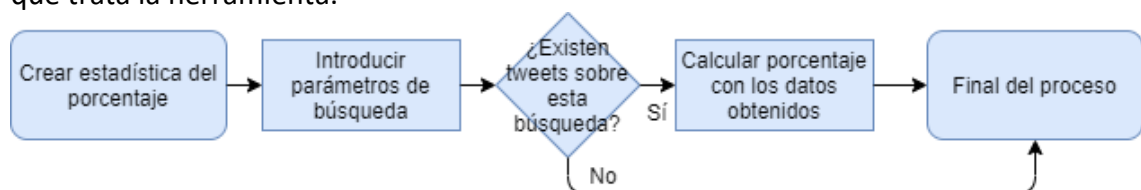


Ilustración 9. Diagrama crear estadística del porcentaje.

La herramienta genera una media aritmética de la polaridad sobre cada día en los que se han recopilado los tuits. Se recogen los tweets de la base de datos pertenecientes a la búsqueda que realiza el usuario y realiza la media para cada día. La búsqueda realizada por el usuario puede ser de un tema y realizarla sobre todas las categorías o de un tema y solo sobre una categoría.

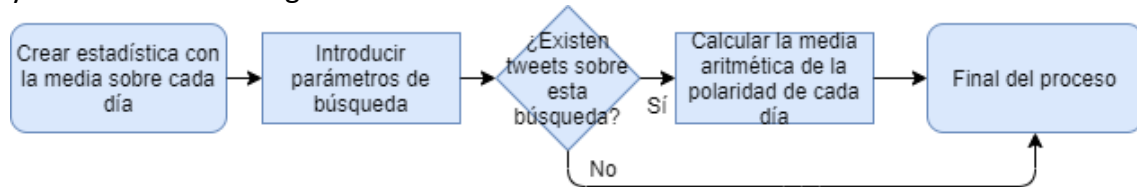


Ilustración 10. Crear estadísticas sobre la media.

➤ Leer estadística

El usuario selecciona los parámetros necesario para poder observar la página que muestra las estadísticas sobre el tema que desee. Introduce la clave del tema, la categoría y el tipo de búsqueda. El sistema recoge las estadísticas generadas y las muestra.

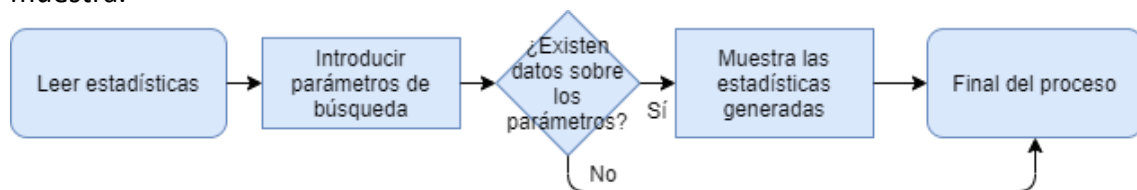


Ilustración 11. Leer estadísticas.

4

Herramientas para el desarrollo

4.1. Colección de herramientas

A lo largo de este capítulo se describen las herramientas que se han estudiado para la realización de este proyecto. Primero, se estudiará que es una API REST debido a que se usará alguna y hay que entender de qué trata. Seguidamente, se observarán las distintas API que se puedan utilizar, como por ejemplo, la primera que se llevará a cabo para el desarrollo, es la que permita la recopilación de tuits. Acto seguido, el siguiente objeto sobre el que se trabaja es el análisis de sentimientos, con lo que, es necesario realizar un estudio de las API que se encuentran en el mercado para hacer el mejor uso de las mismas. Una vez mostrado todas las relevantes, se seleccionará la más conveniente para la herramienta que se está desarrollando. [12] [13]

4.2. Concepto de API

Para el desarrollo de la aplicación es necesario la definición y documentación de los contratos entre aplicaciones usando técnicas de conexión. Estas son las que permitirán realizar la conexión entre nuestra herramienta y la red social Twitter. Estos contratos reciben el nombre de Interfaces de Programación de Aplicaciones o API.

Cuando los programas acceden a servicios proporcionados por otros se utiliza el planteamiento conocido como Arquitectura Orientada a Servicios o SOA. Un planteamiento SOA es aquel en el que una aplicación principal usa los servicios de otras aplicaciones. En el caso de este proyecto, la herramienta hace uso de los servicios que se obtienen a través de la aplicación que ofrece Twitter para la recolección de tuits para su posterior análisis.

API REST fue definida por Roy Fielding en su tesis doctoral *Architectural Styles and the Design of Network-based Software Architectures* [14]. El servicio de transferencia de estado representacional o REST no es una arquitectura software como tal, sino que es un conjunto de restricciones con las que se puede crear un estilo de arquitectura para poder desarrollar una aplicación web a través de HTTP. Dentro del mundo empresarial, muchas usan este tipo de API, y como no, la red social con la que se va a trabajar durante este proyecto la usa.

Las funciones se realizan de manera similar a la web. El cliente y el servidor están conectados débilmente. El cliente no necesita conocer con detalle la implementación del servidor y el servidor no conoce cómo usa los datos el cliente. No se establecen sesiones para la realización de conexiones, con lo que, no tienen estado. La infraestructura de red debe soportar una caché de varios niveles para evitar repetir conexiones repetidas.

Para hacer uso de las características de la aplicación a la que se realiza la conexión, se utilizan las diferentes operaciones:

- GET: Consiste en obtener la información del servidor. Esto puede realizarse a través de un navegador.
- POST: Consiste en enviar información desde el cliente para que sea procesada y actualice o agregue información en el servidor. Esta operación no se puede realizar mediante un navegador.
- PUT: Esta operación permite la edición de los recursos que ofrece el sistema a la que se conecta.
- DELETE: Consiste en la eliminación de los recursos ofrecidos por la herramienta.

Para hacer uso de las API se debe conocer el método de uso que el usuario que la ha creado le proporcionó. Para ello, existen diferentes métodos, uno de ellos es el uso de claves o como se suele conocer como API Key. Esto permite conocer quién está haciendo uso de los servicios que ofrece y cómo se está haciendo. Existen distintos niveles de servicio, estos pueden ser gratuitos o de pago, políticas que limitan el número de peticiones que un usuario puede realizar durante un determinado periodo de tiempo o sin ninguna privatización. Como se ha comentado anteriormente, para hacer uso de las API a veces hay que registrarse o autenticarse. Se conocen las siguientes:

- API Key: Es una cadena larga de caracteres que se asigna a un consumidor de una API cuyo valor es único y se utiliza por parte del usuario en cada una de las solicitudes a la API.
- Open Authorization: También conocida como OAuth, es un protocolo que permite la autorización segura de una API. Se basa en el uso de token de acceso, conocida como access token. OAuth es un framework que permite la delegación de la autorización de acceso a la API. Estos tokens de acceso no contienen identificación del usuario que la está utilizando. Este método es el que utiliza la empresa Twitter.

4.3. API Twitter

Como se ha estado señalando durante el apartado de la explicación sobre que tratan las API, para acceder a la información que se encuentra en Twitter se realiza mediante la suya. Podemos destacar las que poseen:

- Streaming API: Es una API que permite al usuario la monitorización y descarga de los tuits o de la información a tiempo real que está apareciendo.
- REST API: Permite la descarga de los tuits que se han introducido en el tiempo en la red social. A demás de esto, permite que el usuario actualice su estado, lea información de otros usuarios y otras actividades.
- Search API: Suministra los tuits con una profundidad en el tiempo de siete días que se ajustan a la consulta solicitada, con la posibilidad de realizar el filtro por lenguaje y localización.

Para hacer uso de las API, hay que tener en cuenta la seguridad que le sigue. En este caso, se tiene que pedir el acceso a la aplicación. Para ello, es necesario la obtención de una cuenta de Twitter, a continuación, se introducirá en el apartado de desarrollador para la creación de una aplicación y en ese momento se mostrarán la claves asociadas al usuario. Se observa que la autenticación a la API es sobre un OAuth, con lo que, aparecerán los siguientes tokens de acceso:

- Consumer Key (API key)
- Consumer Secret (API secret)
- Access Token
- Access Token Secret

Las dos primeras claves que se obtienen para la aplicación permiten realizar peticiones a Twitter y las dos últimas son las que permiten las solicitudes a las API en nombre de la cuenta del usuario.

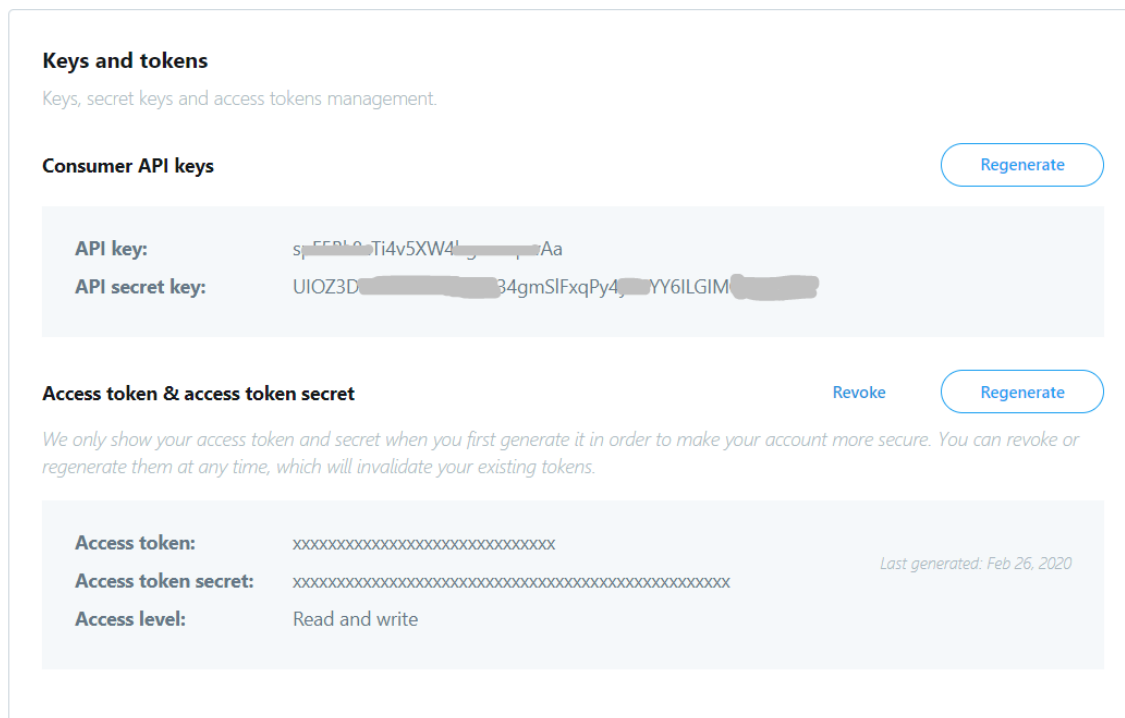


Imagen 7. Token de la aplicación de Twitter.

Una vez conocidos los tokens que permiten hacer uso de la aplicación que se ha creado para el uso de la API de Twitter, es hora de conocer las limitaciones de esta.

- REST API: La limitación que existe es variable dependiendo del método que se haya solicitado. La restricción se basa en solicitudes que se pueden realizar durante 15 minutos. Se localizan diferentes peticiones que a su vez registran otras peticiones, lo que permite una agilización de la descarga.
- Streaming API: Al realizarse mediante una petición a hora real, se denomina que es un flujo continuo y la restricción que se aplica es al caudal recolectado, su máximo es de 50 tuits por segundo.
- Search API: La limitación son de 180 peticiones cada 15 minutos.

Para una mayor visualización de las limitaciones se muestra en la tabla 2 las principales peticiones de las API mencionadas y la limitación asociada a cada una de ellas.

API	Petición	Máximo de datos	Cada 15 minutos
REST	GET user_timeline	200 Tuits	900*200 = 180.000 Tuits
Search	GET search/tweets	100 Tuits	180*100 = 18.000 Tuits
Streaming	POST statuses_filter	-	Máximo 45.000 Tuits

Tabla 1. Limitaciones API Twitter.

4.4. Herramientas de análisis de sentimientos

A lo largo del siguiente apartado se muestran las diferentes API que existen en el mundo empresarial, una definición y caracterización de cada una de ellas para conocer cuál es la más eficiente teniendo en cuenta la herramienta a desarrollar.

- **Sentiment Analysis API - Meaningcloud**

La API de análisis de sentimientos de MeaningCloud proporciona una solución con un análisis detallado de sentimientos multilingüaje de textos de diferentes fuentes. Identifica la polaridad positiva, negativa y neutral que contiene cualquier texto. Para ello, identifica la polaridad local de las diferentes oraciones en el texto y se evalúa la relación que existe entre ellas. El usuario puede crear el propio diccionario y detectar el sentimiento de los elementos incluidos y la definición de un propio modelo de opinión. [15]



Imagen 8. Logo MeaningCloud.

- **MonkeyLearn**

MonkeyLearn ofrece una API para el análisis de sentimientos a través de Machine Learning. Analiza los textos para extraer las palabras claves sobre temas, sentimientos e interacción. Ofrece la modificación de las necesidades para que los usuarios utilicen de forma efectiva el sistema. [16]

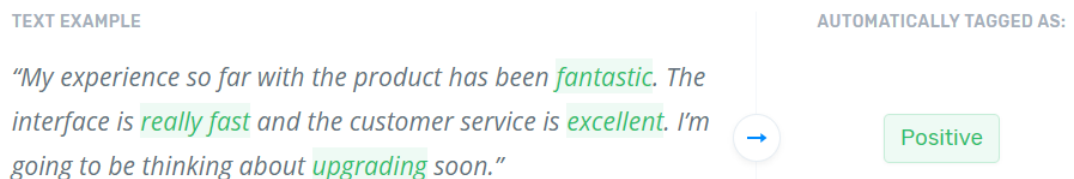


Imagen 9. Ejemplo de análisis MonkeyLearn.

- **IBM Watson Natural Language Understanding**

Permite el análisis de sentimientos sobre textos realizando un análisis global del conjunto de datos que obtiene. Los idiomas sobre los cuales puede realizar el análisis son inglés, español, alemán. Permite la polarización de los textos. [17]



Imagen 10. Logo IBM Watson.

- **Indico**

La API de Indico realiza un análisis de sentimientos estableciendo una caracterización numérica global del texto. Esta evalúa su polaridad dependiendo del valor resultante de la evaluación. Permite el análisis en diferentes idiomas, entre los cuales cabe destacar el español, inglés y alemán. [18]



Imagen 11. Logo Indico.

- **AYLIEN**

Realiza un análisis de opinión a nivel de documento a través de un mensaje global del mismo, sea un texto largo o corto. Está basado en aspectos y a nivel de entidad para la comprensión del contenido de las noticias. Permite el desarrollo de análisis en varios idiomas como el inglés, español, alemán, francés. Pero solo permite la clasificación en inglés. [19]



Imagen 12. Logo Aylien.

- **TheySay PreCeive REST API**

Este servicio permite el análisis de sentimientos en textos tanto en inglés, como en alemán y español. Aparte de realizar una clasificación de polaridades, también permite la detección de emociones que se registran en el texto introducido. Véase un ejemplo, en la imagen 13, que permite la página web para observar los distintos usos que puede realizar. [20]

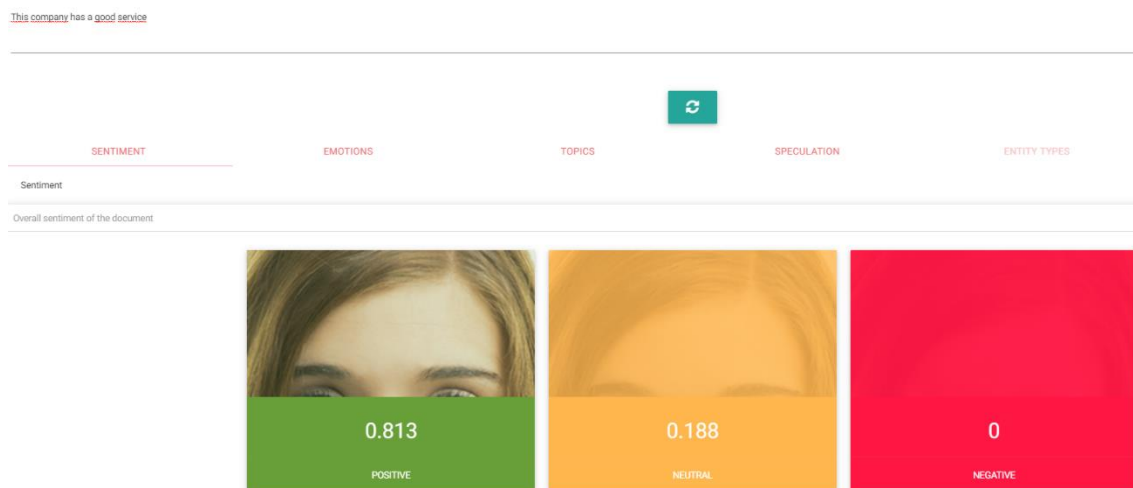


Imagen 13. Ejemplo de TheySay.

- **Google Cloud Natural Language API**

La solución de Google para el análisis de sentimientos actúa como las demás aplicaciones que se han tratado, realiza una clasificación de polaridad sobre el texto enviado por el usuario. También permite un análisis a nivel de documento como pueden ser PDF, archivos de texto plano. Permite el entrenamiento de propios modelos de análisis. [21]

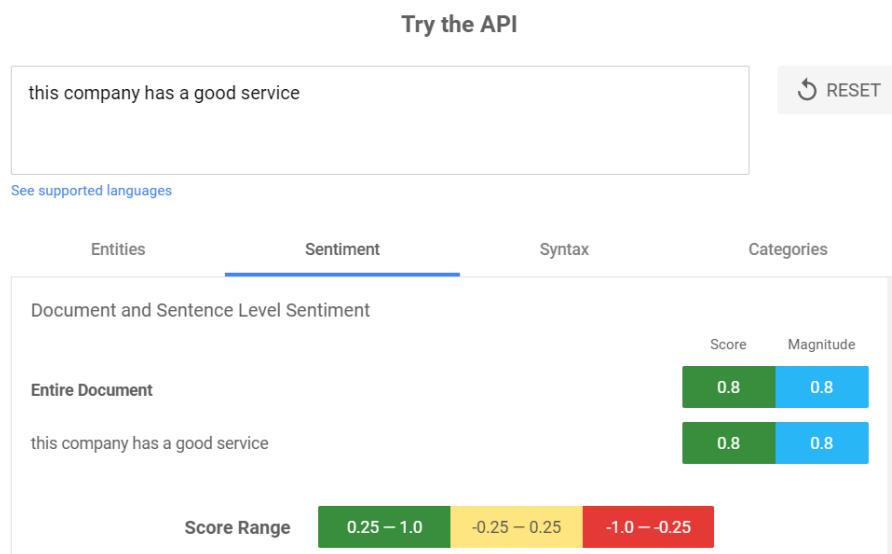


Imagen 14. Ejemplo API Google.

- **Microsoft Azure Text Analytics API**

A través del análisis que permite la solución de Microsoft para el análisis de sentimientos devuelve puntuación y etiquetas a nivel de documento y oración. Esta puntuación, como en el resto de soluciones, devuelve la polaridad del texto introducido. A nivel de documento, permite la devolución de opinión mixta sin puntuación. Permite idiomas como el inglés, alemán, español, francés. [22]

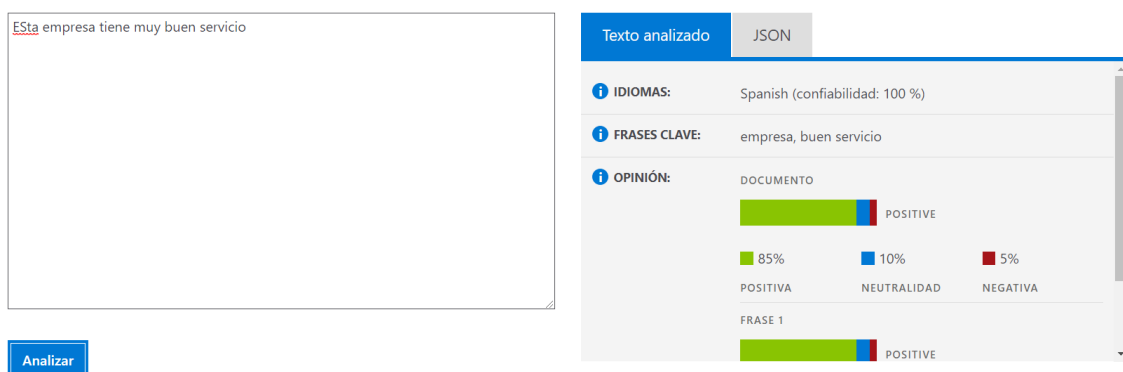


Imagen 15. Ejemplo de API Microsoft.

- **TextBlob.**

TextBlob es una biblioteca de Python para el procesamiento de datos textuales. Realiza un procesamiento del lenguaje natural, de entre los cuales destaca el análisis de sentimientos. Cuando se realiza el análisis, devuelve una tupla donde se encuentra el sentimiento y la subjetividad de la frase. Esto tendrá un rango entre -1 y 1 para conocer si es positivo o negativo el sentimiento, y entre 0 y 1 para medir la subjetividad. Permite una traducción entre idiomas, de los cuales, se puede encontrar el español. [23]



Imagen 16. Logo TextBlob.

- **VADER**

Es una herramienta de análisis de sentimientos basadas en reglas y léxico. Utiliza una combinación de palabras que se etiquetan de acuerdo con la semántica positiva o negativa, esto permite la clasificación de polaridad del texto a analizar y nos permite conocer cuán positivo o negativo es el sentimiento. Solo permite el procesamiento en inglés. [24]

- **Sentiment140**

Sentiment140 es una web que permite el análisis de sentimientos de tweets, para hacer uso de ella es necesario iniciar sesión en Twitter e iniciar el análisis. Uno de los aspectos que destaca es el uso de su API para la clasificación de tweets, permite la integración de esta API en cualquier aplicación. Cabe destacar que permite realizar el análisis en español y muestra la polaridad del texto del tweet. [25]

4.5. Herramientas para el sistema

Una vez vistas las herramientas y API encontradas para la recopilación de los tuits y para el análisis de sentimientos necesarios para la herramienta a desarrollar, es el momento de seleccionar cuáles de ellas se van a utilizar en este trabajo.

Para la recopilación de tuits se usará una API denominada *Tweepy* [26]. Es una librería que permite la conexión con la API de Twitter de una manera sencilla desde Python. Para acceder al contenido que ofrece la API, es necesario el uso de credenciales que ofrece Twitter para desarrolladores. Una vez conseguido las credenciales, se deben de introducir las claves para la autenticación, tal como se observa en la documentación de Tweepy.

Para la búsqueda de tuits, la documentación indica la utilización de varias funciones como son `api.cursor()` y `api.search()`. Estas dos funciones permite la recopilación de tuits

introduciendo varios parámetros como el número de tuits que se quiere recoger, la fecha en la que se quiere empezar la recopilación. También se encuentra la función `api.stream()` que permite la recopilación en tiempo real de los tuits que se están publicando, gracias al parámetro `track` se puede indicar sobre qué temática se está recopilando la información.

Estas funciones serán utilizadas en la herramienta debido a que se realizará una colección de los tuits que se están publicando actualmente para conocer las opiniones más recientes y, además, se requiere recoger la información que se conocía anteriormente para realizar la comparación.

Para la realización del análisis de sentimientos se usarán dos herramientas que se han visto en la sección anterior, la API que ofrece MeaningCloud y la biblioteca que ofrece Python con TextBlob.

MeaningCloud realiza el análisis de sentimientos tanto en inglés como en español, con lo que su uso se realizará sobre texto directamente en español. El análisis que realiza devuelve el siguiente ratio:

Ratio	Significado
N+	Muy negativo
N	Negativo
None	Neutral
P	Positivo
P+	Muy Positivo

Tabla 2. Ratio de MeaningCloud.

El problema que se obtiene con esta API son las limitaciones en la versión gratuita. En la web aparecen las peticiones o créditos usados, esta permite el uso de 20.000 peticiones al mes. Debido a esta restricción, el uso que le daremos a esta API es para un uso simple para no consumir todos los créditos: análisis simples de pocos textos.

TextBlob es una biblioteca que realiza el análisis de sentimientos en inglés, pero tiene una función que permite la traducción de textos. Con lo cual, se recogerá el texto que se desea analizar y se realizará la traducción que ofrece la biblioteca y así pudiendo realizar dicho análisis. Este será el método más común que se utilizará en la aplicación debido a que no hay restricciones. Devuelve el siguiente ratio de análisis:

Ratio	Significado
-1	Muy negativo
0	Neutral
1	Muy positivo

Tabla 3. Ratio de TextBlob.

Para la utilización de estos dos métodos, es necesario poner en común estos ratios que devuelve para poder conocer la polaridad de los tuits. Para ello, se ha escogido como mejor opción, optar por el ratio que ofrece TextBlob como puesta en común, debido a que usa la diferencia sobre tres polaridades.

4.6. Utilidades para el desarrollo

Como se ha explicado anteriormente en las herramientas para el sistema, se conoce que el lenguaje que se va a utilizar va a ser Python. [27]



Imagen 17. Logo Python.

Python es un lenguaje multiparadigma de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa o páginas web. Es interactivo, lo que quiere decir que posee un intérprete por línea de comandos con lo que se pueden introducir sentencias. Cada una de ellas se ejecuta y produce un resultado visible que permite un mayor entendimiento del mismo. Uno de los aspectos fundamentales es la disponibilidad de funciones incorporadas en el propio lenguaje, para el tratamiento de cadenas de caracteres, números, archivos. Además de esto, también posee una cantidad de librerías que se pueden importar a los programas para tratar los temas diversos. Cabe destacar que posee una sintaxis muy visual caracterizada por la indentación de los códigos, esto se refleja mediante los márgenes en el código. Como uno de los objetivos para la herramienta era poder utilizarla a través de la web y como se va a utilizar Python para la recolección de los tweets y el análisis de sentimientos, será necesario también hacer uso de un framework para la web. Un framework web es un conjunto de componentes que ayudan al desarrollo de sitios web más fácilmente. El que se va a utilizar es Django. [28]



Imagen 18. Logo de Django.

Django es de código abierto y está escrito en Python. Es muy rápido, se puede construir una aplicación en poco tiempo. Hay una gran comunidad en Django, por lo que si se necesita realizar algo, lo más habitual es que ya este implementado, con lo que, solo habría que adaptarlo al uso personal. También es muy escalable, permite pasar a un sistema complejo en poco tiempo.

En este proyecto para realizar el análisis que se quiere hay que recopilar un volumen de información muy grade, con lo que, habría que tener algún sistema para almacenarla. Para esto se ha elegido MongoDB. [29]



Imagen 19. Logo de MongoDB.

MongoDB es una base de datos del tipo NoSQL, acrónimo de No solo SQL, de documentos, los cuales son puros documentos JSON, que ofrece una gran escalabilidad y flexibilidad con un modelo de consultas e indexación avanzado. Tiene la capacidad de ser muy veloz, esto permite que en poco tiempo almacene los datos que se recolectan con la herramienta gracias al almacenamiento de los documentos. También posee la característica de almacenar y administrar volúmenes gigantescos de datos en sus entidades.

Para el desarrollo de la herramienta se va a utilizar el IDE PyCharm. IDE o Entorno de Desarrollo Integrado es una aplicación destinada a brindar servicios integrales que permite la construcción de código de una manera más sencilla.



Imagen 20. Logo de PyCharm.

PyCharm [30] es uno de los entornos de desarrollo más completos para Python. Es una de las herramientas que ofrece JetBrains [31]. Esta aplicación tiene un costo anual, pero gracias a Github estudiante, ofrece la posibilidad de tener una versión gratuita para estudiante con todos los privilegios. Posee un editor inteligente que permite el autocompletado de código con algunos atajos que posee en el teclado. Además permite realizar una simple navegación entre las diferentes clases que se han ido creando.

5

Diseño de la herramienta

5.1. Arquitectura de la herramienta

En este apartado se encuentra la arquitectura de la herramienta. Esta define la forma en la que el sistema se estructura, de qué módulos se compone, etcétera.

Para la elaboración de la herramienta se necesita la conexión entre los diferentes módulos que la componen. Este sistema está estructurado en las siguientes partes.

- Recopilador de tuits: Este será la parte encargada de recopilar los tuits de la aplicación de Twitter. Realizará una limpieza de los datos mientras se estén recogiendo información.
- Analizador de sentimientos: Se encarga de realizar el análisis de sentimientos de la información obtenida.
- Base de datos: Parte en la que se almacenan los datos que se obtienen con las diferentes características que se requieren para la aplicación.
- Back end de la Web: Es la lógica de la aplicación, la cual se conecta con la parte de recopilación de información.
- Front end de la Web: Es la parte que el usuario observa e interactúa para hacer uso de la herramienta.

El Back-end es la parte del sistema que conecta el front-end utilizado para que el usuario interactúe con el recopilador de tuits. A través del front-end, se podrán seleccionar las temáticas y las categorías de la información que se quiere recopilar, y esta petición se enviará al back-end. Una vez encontrados los tuits que se han pedido desde el back-end, estos serán analizados en la parte del analizador de sentimientos para conocer la

polaridad de los datos. A continuación, el sistema guardará en la base de datos la información de los tuits y será accesible a través del back-end de la aplicación web.

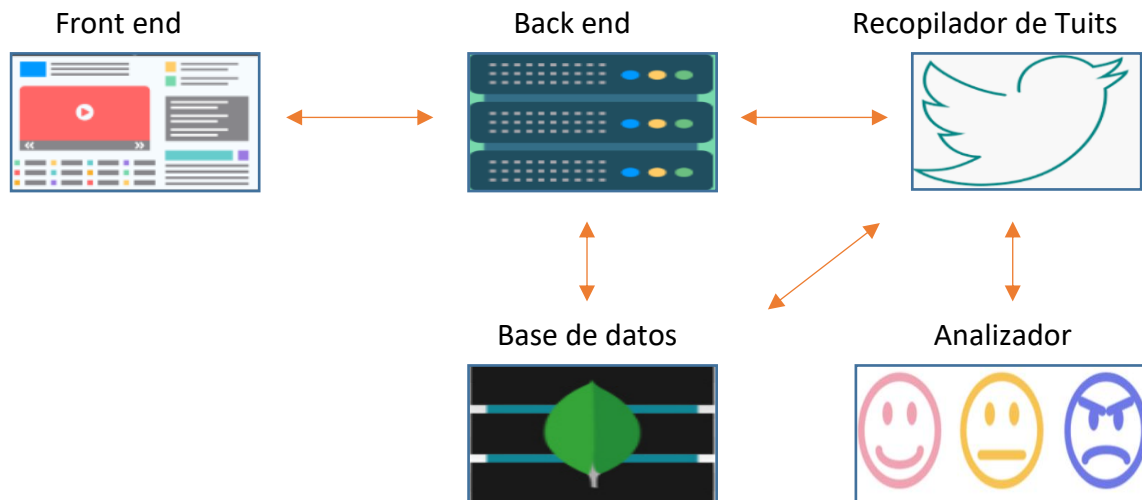


Ilustración 12. Esquema de la arquitectura.

5.2. Diseño de la web

Uno de los aspectos para el uso de la herramienta es el desarrollo de una aplicación web, donde el usuario puede interactuar con ella. Para ello, en este punto se muestra el diseño que se desea que tome. Ha de contener los apartados necesarios para que cumplan con todos los objetivos planteados y, además, información que debe de contener todo sistema de información en internet, como la información del usuario y para qué sirve la herramienta.

Se comienza con el punto inicial de la aplicación o lo que se suele conocer como el inicio principal o índice. El diseño que este debe tener es una navegación que permita al usuario conocer de primera mano el uso de la herramienta, información del usuario que la ha desarrollado y poder ir al apartado que contiene el recopilador de tuits y su analizador.

El inicio de la web contiene una breve descripción la funcionalidad de la herramienta que se ha desarrollado y su logo. La estructura que siguen las diferentes ventanas consiste en un contenedor en el que en la parte superior se tiene los diferentes enlaces hacia los apartados que se han destacado anteriormente y en la posición media se muestra el contenido del apartado. Se observa en la Imagen 21.



Imagen 21. Inicio de la web.

En el apartado sobre mí, como se aprecia en la Imagen 22, aparece una descripción de quién ha desarrollado la aplicación, así como, varias formas de contactar con el usuario y una imagen sobre la persona. También se encuentran los logos de la Universidad de Málaga y la Escuela Técnica Superior de Ingeniería Informática.



Imagen 22. Sobre mí de la web.

En la parte del analizador se puede ver los temas que se hayan creado con anterioridad en la parte izquierda de la página, junto con un botón para una posible modificación del tema. En la parte de la derecha, se observa una breve explicación de lo que se trata cuando se está hablando de tema. En la parte inferior, se observa un botón que será el indicado para la creación de los temas de búsqueda. Obsérvese la Imagen 23.

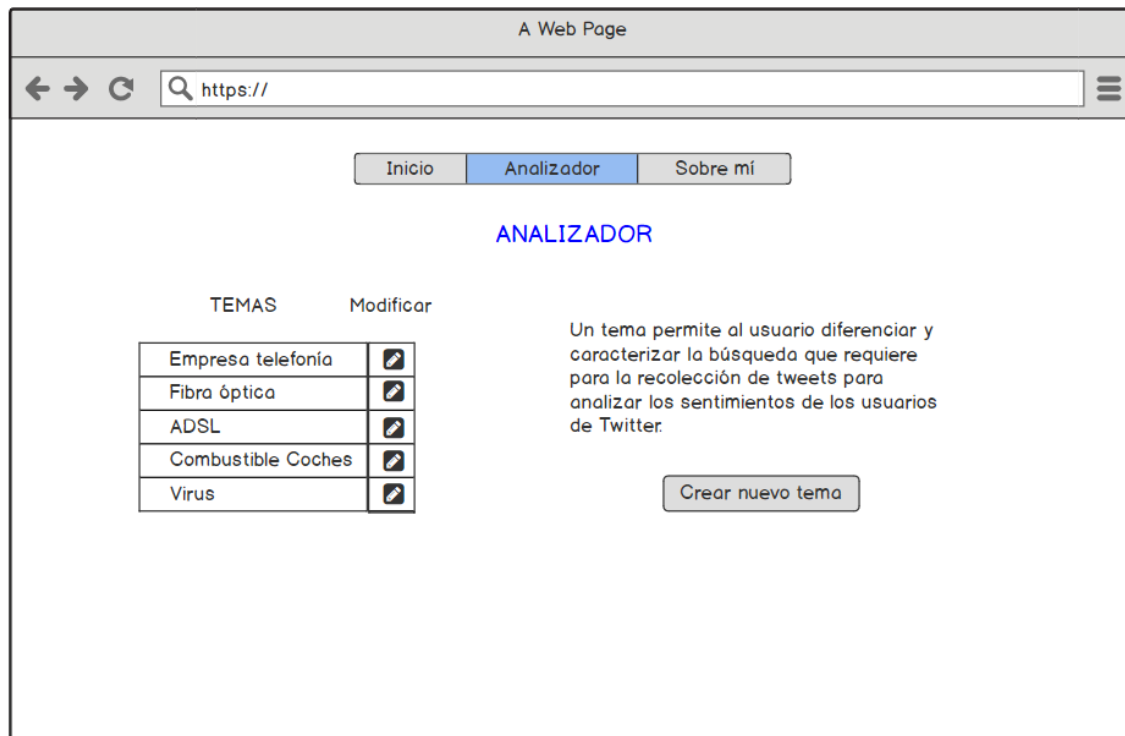


Imagen 23. Muestras de temas en la web.

En el apartado de la creación de temas, pedirá que el usuario inserte el nombre del tema, palabras clave para realizar la búsqueda del tema y sus categorías.



Imagen 24. Creación de tema.

La siguiente pestaña a destacar es la visualización de las categorías. En la parte de la izquierda se observan las diferentes categorías que se han obtenido para el tema que se ha seleccionado. En la parte de la derecha aparece una explicación sobre qué es una categoría y para crear una nueva. También se mostrará un análisis sobre todas las categorías en general, tanto histórico, como actual o ambas opciones.

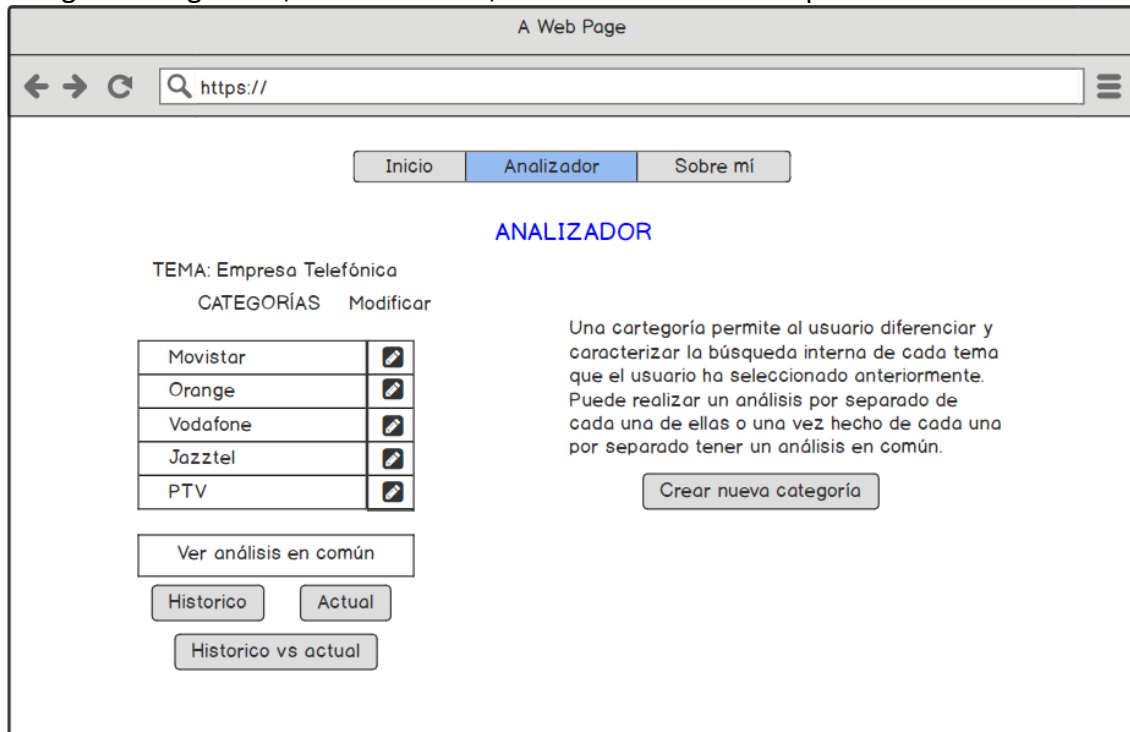


Imagen 25. Visualización de categoría.

Para crear la categoría, hay que introducir su nombre y las palabras clave para realizar la búsqueda.



Imagen 26. Creación de categoría.

Se selecciona una categoría en la cual aparecerán las opciones para la recopilación de los tuits. Como se observa en la Imagen 27, aparece un apartado donde se pueden realizar búsquedas históricas para conocer los datos que existían anteriormente y así para poder hacer una comparación con la actualidad, también ofrece poder realizar esa búsqueda con los datos actuales que se irán recogiendo a medida que los usuarios vayan publicando en Twitter. Aquí permite la visualización de las estadísticas que se han creado con los datos ya recogidos diferenciando los actuales o los históricos, y ofreciendo la posibilidad de visualizarlos juntos para ver si ha habido mejoría en la opinión que se tiene sobre dicha categoría.

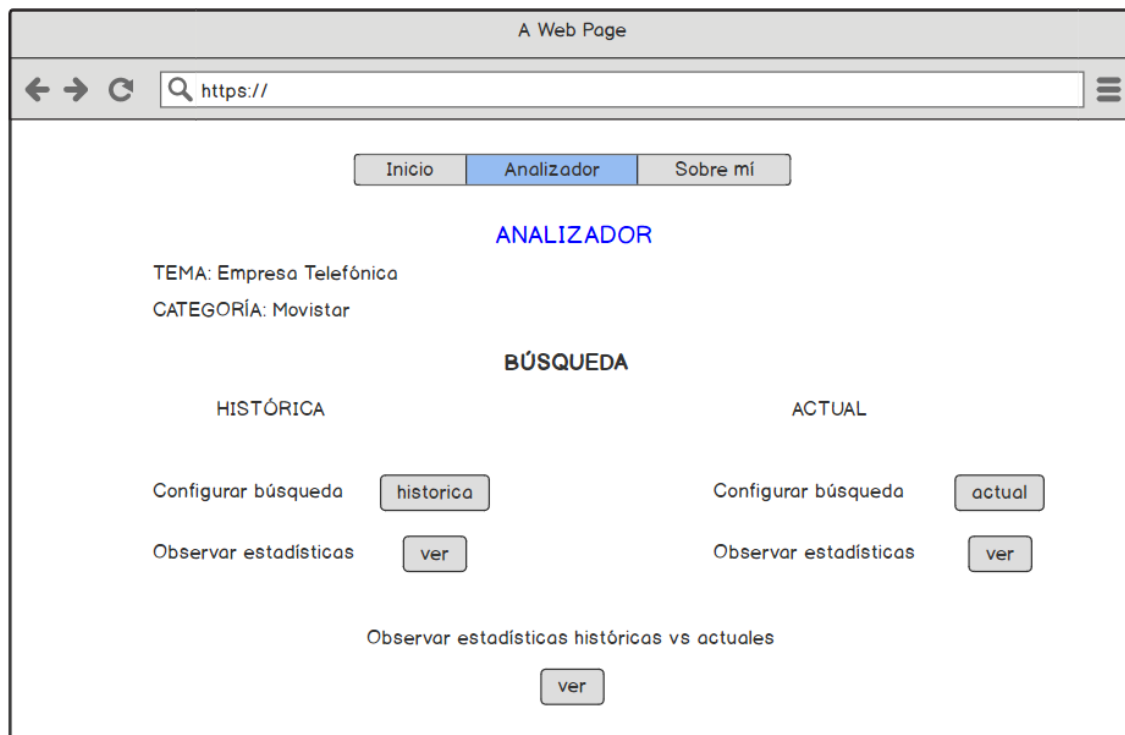


Imagen 27. Búsquedas de categorías.

Para la configuraciones de búsqueda que se puede realizar en la herramienta, se ha de diferenciar la que permite la recopilación de tuits históricos que hay en la red social y la que permite la recolección de las publicaciones que se están publicando desde que se inicia la búsqueda. Como ambas configuraciones son similares, solo se pondrá una representación mostrando las diferencias que se encuentran entre ambas. El primer caso que se observa es que para la recopilación histórica, hay que introducir dos fechas, una que indica la fecha del primer tuit que se quiera recoger y la segunda indica la fecha del último tuit. En el caso de la actual, solo aparecerá una fecha que es la que indica hasta cuándo se quiere realizar la recopilación. También aparecerá una opción para introducir el número de tuits que se quieren almacenar y analizar. Como se ha comentado anteriormente durante el desarrollo de este documento, se utilizan dos formas para el análisis de sentimientos: uno referente a la biblioteca TextBlob que no tiene limitaciones y otro usando la API de MeaningCloud que tiene limitaciones, aquí se puede seleccionar cuál de ellas se va a utilizar, pero se señala que para hacer uso de MeaningCloud, el número de tuits ha de ser inferior a la cuota que permite la API.

A Web Page

← → ↻

Inicio Analizador Sobre mí

ANALIZADOR

TEMA: Empresa Telefónica
CATEGORÍA: Movistar

CONFIGURACIÓN DE BÚSQUEDA

HISTÓRICA		ACTUAL	
Desde	<input type="text" value="/ /"/>	Hasta	<input type="text" value="/ /"/>
Hasta	<input type="text" value="/ /"/>	Número de tweets	<input type="text" value="NUM"/>
Número de tweets	<input type="text" value="NUM"/>	Analizador	<div>TextBlob ▼ Meaning</div>
Analizador	<div>TextBlob ▼ Meaning</div>	<div>crear</div> <div>cancelar</div>	

Imagen 28. Unión del diseño configurar búsqueda actual e histórica.

Una vez vista todas las posibilidades que hay para la recopilación de tuits, es hora de conocer la visualización de las estadísticas. En este apartado se muestra una representación sobre cómo se muestran todas las estadísticas, debido a que es algo que tienen en común muchos apartados, solo aparecerá una imagen que lo represente. Se podrá observar el número total de tuits que se han recolectado, información con el número de tuits positivos, negativos y neutros y varios gráficos comparativos. Cuando se esté visualizando la comparativa entre históricos y actuales, se observarán ambas estadísticas en la misma pantalla. Véase la Imagen 29.

A Web Page

← → ↻

Inicio Analizador Sobre mí

ANALIZADOR

TEMA: Empresa Telefónica
CATEGORÍA: Movistar

Visualización de estadísticas

ACTUAL		Visualización de estadísticas	
Desde	<input type="text" value="/ /"/>	Número de positivos	<input type="text" value="NUM"/>
Hasta	<input type="text" value="/ /"/>	Número de neutros	<input type="text" value="NUM"/>
Número de tweets	<input type="text" value="NUM"/>	Número de negativos	<input type="text" value="NUM"/>
Analizador	<div>TextBlob</div>	 Visualización general	

Imagen 29. Estadísticas generadas.

5.3. Diseño de la base de datos

Para el almacenamiento de los datos se va a utilizar MongoDB y, como se ha comentado anteriormente, es una base de datos de tipo no relacional, con lo que, los elementos que la componen son documentos. Los tipos que se diferencian en la herramienta son los diferentes requisitos de información que se usan en el sistema. A continuación se muestran los diferentes objetos que la componen y sus características:

➤ Tema

El tema referencia la búsqueda que se realiza para la recopilación de los tuits que desea el usuario. Para ello, es necesario la obtención del nombre y de las palabras clave que se usarán en la búsqueda. Dentro de este objeto, aparecen las diferentes categorías que tiene el tema, con la información relevante del nombre y las palabras clave, que como en el caso del tema, sirven para realizar la búsqueda. La estructura se muestra a continuación:

Elementos del TEMA	Elementos de la categoría	Definición
Nombre		Nombre del tema
Palabras_Calve		Palabras clave para la búsqueda
Categorías		Array que contiene las categorías
	Nombre_categoria	Nombre de la categoría
	Palabra_clave_categoria	Palabras clave de la categoría

Tabla 4. Elemento Tema de la base de datos.

➤ Tweet

A la hora de llevar a cabo la recolección se obtiene el objeto tuit en formato JSON con información referente a Twitter. Se observa que hay información que no se necesita para el estudio del análisis de sentimientos, con lo que, se seleccionan elementos que se ha considerado como necesarios para ello. A continuación, se describirán los datos del objeto recopilado que se utilizarán. Posteriormente, se realizará el estudio sobre qué significan los diferentes apartados que se van a recoger. Por último, se mostrará la representación que tendrá el objeto tweet en la base de datos del sistema.

• Elementos obtenidos de Twitter

A continuación se muestra una tabla con los elementos que se obtienen del objeto tuit a través de Twitter con la identificación y el significado que refleja cada uno de ellos.

Nombre	Significado
Created_at	Fecha de creación del tuit
Id	Identificador del tuit
Full_text	Contenido del texto del tuit
Truncated	Si ha sido recortado el texto
geo	Lugar de geolocalización
Coordinates	Coordenadas de la localización
Place	Lugar de localización
Retweet_count	Número de retuit que tiene

Favourite_count	Número de favoritos que tiene
Favourited	Si el tweet es un favorito o ha sido publicado de cero
Retweeted	Si el tweet es un retuit a otro o ha sido publicado de cero
lang	lenguaje en el que se ha escrito

Tabla 5. Información que ofrece Twitter referente a un tuit.

- Elementos para la base de datos

A continuación, se muestra una tabla en la que se relaciona cada elemento que pertenece al objeto tweet de la base de datos con los elementos que se han observado que ofrece la red social.

Tweet Base de datos	Tweet de Twitter
Fecha_creado	Created_at
Id_twitter	Id
Texto	Full_text
Truncado	Truncated
geo	geo
Coordenadas	Coordinates
place	Place
Número_ret	Retweet_count
Número_fav	Favourite_count
EsFavorito	Favourited
EsRetweet	Retweeted
Idioma	lang

Tabla 6. Comparación de los objetos tweet.

A parte de estos datos, también se guardan los objetos referentes al análisis que se han realizado sobre los sentimientos y sobre los temas que tratan los tuits con sus categorías.

Elemento del tweet	Significado
Tema	Nombre del tema al que pertenece
Categoría	Nombre de la categoría al que pertenece
Polaridad	Contiene la polaridad del tuit
Número_Polaridad	La polaridad del tuit representada con un número
Búsqueda	Si es en una búsqueda actual o histórica
Análisis	Si se ha realizado el análisis con la biblioteca TextBlob o MeaningCloud

Tabla 7. Elementos que se añaden al objeto tweet.

5.4. Realización de las estadísticas

Como se ha observado durante el desarrollo del documento, uno de los aspectos que se ha señalado es la generación de estadísticas sobre los temas y las categorías que el usuario ha creado. Para esto, es necesario que el objeto tweet de la base de datos de la herramienta tenga un elemento en el que guarde la información referente a la polaridad, el tema y la categoría a la que pertenece.

Para cada categoría se quiere conocer el número de tuits que se han recopilado. Para ello, en la base de datos se puede realizar una búsqueda sobre los elementos que contengan el nombre de la categoría. Como en el caso anterior, ocurre exactamente lo mismo si se quiere conocer esa información referente al tema.

El número total de tuits es necesario para poder elaborar un gráfico donde se observe el porcentaje de opiniones positivas, negativas y neutras que se han encontrado.

6

Implementación

6.1. Inicio de la implementación

En este capítulo se encuentra la información relativa a la implementación que se ha realizado de la herramienta. Como paso previo, se han instalado los siguientes entornos y herramientas:

- Python en su versión 3.8.1; esta versión es la más nueva de este lenguaje de programación, su fecha de lanzamiento fue el 18 de diciembre de 2019 y está en su versión estable. Contiene nuevas características importantes que no poseen versiones anteriores, como por ejemplo, expresiones de asignación, argumentos posicionales, configuración de iniciación.
- El IDE Pycharm con el entorno virtual de Anaconda, que funciona como un gestor interno de paquetes para la ciencia de datos. Permite la creación de entornos virtuales para que las instalaciones que se realicen para este proyecto queden registradas solo en él y no haya conflictos con otros proyectos.
- Una vez instalado lo anterior, se inicia el entorno de desarrollo y se crea un proyecto Django en su versión 2.2.12. Durante la creación del mismo, se crea la estructura interna de carpetas que caracteriza un proyecto Django. Es necesario que el proyecto indique el intérprete del entorno que se ha creado para el mismo, a través de Pycharm, también se permite la creación del entorno desde el mismo momento que se produce la creación.
- A continuación, se genera una aplicación con el nombre de *herramienta*, es una pequeña librería que representa un aspecto concreto del proyecto, esta app tiene la característica de ser reutilizable para otros proyectos. En ella, se generarán una serie de archivos y carpetas que son utilizadas para el desarrollo.

6.1.1. Estructura de ficheros.

La Imagen 30 muestra la estructura generada para el proyecto.

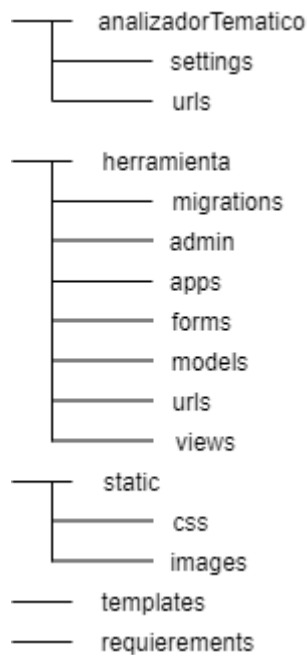


Imagen 30. Estructura de ficheros.

Una vez vista la estructura que sigue el proyecto, se describen a continuación cada uno de sus componentes y el uso que se le ha dado:

- **Analizador Temático:** Encontrándose dentro del mismo los archivos de settings y urls. En settings, como su nombre en inglés indica, se encuentran los ajustes y las configuraciones que posee el proyecto, en él se indica la zona horaria, la base de datos, las aplicaciones instaladas, temas de seguridad, middleware para los ciclos cliente-servidor, entre otras cosas. Urls, contiene las URL principales del proyecto, se crean patrones que deben de seguir las URL de las aplicaciones, entre ellas se encuentran la de administración, una referencia a las urls de la herramienta y una necesaria para iniciar sesión.
- **Herramienta:** Esta carpeta se crea durante el inicio de la aplicación de Django. Migrations contiene las migraciones que se realizan con la base de datos, por ejemplo, cuando se crea la base de datos o cuando se obtiene un cambio en la misma, se debe de realizar una migración y se encuentran en esa carpeta. Admin, registra archivos en el administrador de Python para gestionar de forma gráfica el contenido de la base de datos. Apps son los metadatos de la aplicación. Forms, contiene los formularios que se generan para la aplicación, como por ejemplo, cuando se crea una temática. Models, contiene los modelos de la base de datos, en ella se encuentra el modelo tema, categoría y tweet. Urls, son las URL de la herramienta, se añaden con patrones que siguen una expresión regular y está relacionado directamente con la siguiente. Views, son las vistas de la herramienta, funciones y los controladores actuando en las diferentes páginas desarrolladas.
- **Static:** Una carpeta de archivos estáticos, donde se puede encontrar el CSS de la herramienta para el diseño visual y las imágenes que contiene.

- **Templates:** Aparecen las diferentes plantillas que se utilizan para la herramienta.
- **Requirements:** Un fichero donde se definen el listado de dependencias del proyecto.

6.2. Desarrollo del script

En este apartado se describe la implementación del script que permite la recopilación de los tuits que se publican en la red social y el análisis de sentimientos que se produce para conocer su polaridad. Cabe destacar, que se desarrollan diferentes partes, una que indica la recopilación de tuits de manera actual, usando el procedimiento de transmisión que ofrece la API y la forma histórica, diferenciada en si misma por el uso del cursor de paginación y la búsqueda sin paginación. En la parte del análisis de sentimientos, se encuentran dos operaciones, una para el uso de la biblioteca de Python llamada TextBlob y otra operación para la API que ofrece MeaningCloud.

6.2.1. Recopilación de tuits.

A continuación, se explican las partes que se han implementado para la recopilación de los tuits. Para el comienzo de cualquier búsqueda que se vaya a producir, es necesario la información del tema y la categoría a la que se le va a hacer referencia. Por esta razón, en ambas implementaciones, se introduce el nombre y las palabras clave, tanto del tema, como de la categoría a la que se desea realizar la búsqueda.

- Actual

Como se ha comentado en la parte introductoria, en la sección de la recolección de tuits, una vez incorporado el nombre y las palabras clave, otro aspecto fundamental para este tipo de búsqueda es el conocimiento de la fecha hasta la que se desea. Esta es la que le indica a la herramienta cuándo debe finalizar su búsqueda. Para el uso de la transmisión, es necesaria la creación de una clase que viene definida en la propia documentación de tweepy que puede ser modificada, se puede observar en Código 1. Esta clase ha de heredar de la clase StreamListener que proporciona la API tweepy. Dentro de la clase que se ha proporcionado, se observan dos funciones, una referente al estado y otra al error. En la función del estado, es donde se obtiene el tuit que se está escuchando para recopilarlo. Para poder usar la información que se recoge de Twitter, se convierte en formato JSON para poder acceder con facilidad a su estructura. Una vez hecho esto, lo pasamos sobre la función que procesa el tuit que se verá más adelante. Cabe destacar, que para realizar esto, se utiliza de forma que el programa recoja las excepciones que puedan aparecer en los datos del tuit, debido a que puede aparecer errores de formato. Analizando también la función para los errores, se han modelado todos los principales errores que pueden aparecer durante la recopilación. Los errores contemplados son: error de conexión; error en la autenticación de Twitter que es necesaria para la recopilación y el uso de tweepy; error en la introducción de los parámetros de búsqueda; otros tipos de errores en la autenticación; y el último error general que es debido al sobrepasar los límites de la API. Para este último error, la solución es detener la ejecución de la herramienta durante 15 minutos hasta que vuelva a su normalidad.

```

class MyStreamListener(StreamListener):
    def on_data(self, raw_data):
        try:
            data = json.loads(raw_data)
            procesarTweet(data, 'TextBlob', 'actual', tema_global,
categoria_global, categoriaPal_global, temaPal_global)
            return True
        except BaseException as e:
            print("Error en el dato: %s", format(str(e)))
            return True #True para que siga recolectando

    def on_error(self, status):
        if status == 420:
            return False #Parar por error de conexión
        elif status == 401:
            print('Error autenticación')
            return False
        elif status == 400:
            print('Solicitud incorrecta: Parámetros mal introducidos')
            return False
        elif status == 403:
            print('No hay productos Lab en la autenticación')
            return False
        else:
            print('Se para durante 15 min')
            time.sleep(60 * 15) # Para los limites de la API
        print(status)
        return True

```

Código 1. Clase Listener.

Una vez conocida la clase necesaria, se debe iniciar la transmisión de la información. En la función para ello, el primer paso es la autenticación y el uso de la API. Se realiza una llamada para recoger el objeto de autenticación y a su vez, introducir la clase de escucha que se ha mencionado anteriormente. Con el nombre y las palabras clave tanto del tema como de la categoría, se crea el filtro que se va a utilizar para realizar la recopilación. A través de la función de filtro e indicando el idioma y la cadena de palabras que hemos generado con los datos anteriores, se le indica a la herramienta la búsqueda que ha de realizar. Obsérvese el Código 2.

```

def stream(nombreTema, palabrasClaveT, nombreCate, palabrasClaveC):
    modificarGlobal(nombreTema, nombreCate, palabrasClaveT, palabrasClaveC)
    auth = get_auth('stream')
    twitter_stream = Stream(auth, MyStreamListener())
    palabra = []
    palabra.append(palabrasClave(nombreTema))
    palabra.append(palabrasClave(nombreCate))
    palabra.append(palabrasClave(palabrasClaveT))
    palabra.append(palabrasClave(palabrasClaveC))
    twitter_stream.filter(languages=['es'], track=palabra)

```

Código 2. Función stream.

- Histórica

Para el proceso de búsqueda histórica, la herramienta necesita conocer los parámetros del nombre y palabras clave, como se ha comentado, y además, necesita conocer dos fechas, la de inicio y de fin que indican el periodo que se desea realizar la búsqueda. Una vez introducido, otro de los aspectos que se proporciona es el número de tuits que se desean recopilar. Se conoce que el límite de la API en ocasiones no permite la recopilación en su totalidad de toda la cantidad que el usuario desea, con lo que se soluciona recogiendo el número de tuits resultantes de la primera búsqueda y se calculan los restantes para llegar al número que el usuario desea. Una vez hecho esto, la función realiza otra búsqueda con un nuevo parámetro que indica el último id del tuit recogido para, así, recopilar los tuits a partir de los que ya se han encontrado y así no repetirlos. La búsqueda permite el uso de seleccionar el motor de análisis, con lo que, es un aspecto que también se introduce en la función para realizar el correspondiente análisis. Como se ha destacado en otras ocasiones, la búsqueda histórica permite el uso de dos funciones, search o cursor. Cursor tiene un mayor límite ya que usa el método de paginación que proporciona Twitter, por eso, este tipo es usado cuando se desea recopilar menos de 1000 tweets. La estructura de ambas funciones no se diferencia, con lo que, solo se mostrará una. La función descrita se encuentra en el Código 3.

```

def cursorHist(nombreTema, palabrasClaveT, nombreCate, palabrasClaveC ,
numTw, maquina, fechaInic, fechaFin):

    auth = get_auth('cursor')
    api = tweepy.API(auth, wait_on_rate_limit=True,
wait_on_rate_limit_notify=True)
    palabra = []
    palabra.append(palabrasClave(nombreTema))
    palabra.append(palabrasClave(nombreCate))
    palabra.append(palabrasClave(palabrasClaveT))
    palabra.append(palabrasClave(palabrasClaveC))
    numTw = int(numTw)
    contTweet = 0
    busqueda = numTw
    sinceId = None
    max_id = -1
    while contTweet < busqueda:
        busqueda = busqueda - contTweet
        print(busqueda)
        try:
            if max_id <= 0:
                tweetemp = tweepy.Cursor(api.search, palabra,
since=fechaInic, until=fechaFin, lang='es',
tweet_mode='extended').items(busqueda)
            else:
                tweetemp = tweepy.Cursor(api.search, palabra,
since=fechaInic, until=fechaFin, max_id=str(max_id - 1), lang='es',
tweet_mode='extended').items(busqueda)

            if not tweetemp:
                print('No encontrado')
                break
            for tweet in tweetemp:
                contTweet += 1
                tweJason = tweet._json
                max_id = tweJason['id']
                procesarTweet(tweJason, str(maquina), 'historica',
nombreTema, nombreCate, palabrasClaveC, palabrasClaveT)

        except tweepy.TweepError as e:
            time.sleep(60 * 15) #Para los limites de la API
            print(e.reason)
        except StopIteration:
            break

```

Código 3. Búsqueda histórica.

6.2.2. Análisis de sentimientos.

Para el proceso de análisis, las funciones que anteriormente se han observado, uno de los elementos es la introducción del motor de análisis que se va a utilizar, en esta variable se indica si el análisis es con la biblioteca Textblob o la API de MeaningCloud. Obsérvese la implementación de las funciones para el análisis de sentimientos que sigue la herramienta.

- TextBlob

Para el uso de esta función, el primer paso que se ha tomado, es la traducción del texto que se obtiene del tuit al inglés, debido a que la biblioteca permite el análisis de sentimientos solo en este idioma. Una vez traducido, pasa por el análisis con la función de la biblioteca y así, obteniéndose un valor sobre la polaridad, con este valor, se indicará la polaridad que define al texto analizado.

```
def analisisText(full_text):
    textoTextBlob = TextBlob(full_text)
    try:
        textoIngles = textoTextBlob.translate(to='en')
    except:
        textoIngles = textoTextBlob

    polarity = textoIngles.sentiment.polarity
    if polarity < 0:
        polaridad = "Negativo"
    elif polarity == 0:
        polaridad = "Neutro"
    elif polarity > 0:
        polaridad = "Positivo"

    return (polaridad,polarity)
```

Código 4. Análisis TextBlob.

- MeaningCloud

Es necesario indicar la clave que se obtiene de la API para poder hacer uso de la misma, introducir el idioma sobre el cual se realiza el análisis. Una vez indicado todo esto, se realiza el análisis y se combina con los índices de la herramienta, indicando así tanto el número de polaridad, como la polaridad a la que pertenece.

```
def analisisMeaning(full_text):
    analisis = meaningcloud.SentimentResponse(
        meaningcloud.SentimentRequest(license_key_MC, lang='es',
txt=full_text, txtf='plain').sendReq())
    if analisis.getGlobalScoreTag() == 'N+':
        polaridad = "Negativo"
        polarity = -1
    elif analisis.getGlobalScoreTag() == 'N':
        polaridad = "Negativo"
        polarity = -0.5
    elif analisis.getGlobalScoreTag() == 'P':
        polaridad = "Positivo"
        polarity = 0.5
    elif analisis.getGlobalScoreTag() == 'P+':
        polaridad = "Positivo"
        polarity = 1
    else:
        polaridad = "Neutro"
        polarity = 0
    return (polaridad,polarity)
```

Código 5. Análisis MeaningCloud.

6.3. Base de datos

Para el uso de la base de datos es necesario la conexión con la misma. Esta herramienta utiliza una base de datos de MongoDB, con lo que, es necesario iniciar el servidor de MongoDB para poder empezar a realizar la conexión. Una vez ejecutado el servidor, es necesario instalar en el proyecto Django, con su versión 1.3.1, que permitirá realizar la conexión con Django. A continuación, a través de la aplicación que proporciona Mongo, MongoDB Compass Community, u otra que permita la creación de bases de datos, realizamos la conexión con el host en el que se encuentra alojada la base de datos; en este caso, irá alojada en localhost con el puerto predeterminado 27017. Una vez hecho esto, la aplicación permite la creación de la base de datos que se va a utilizar en la herramienta. Solo se crea la base de datos, no es necesario la creación de los documentos que contendrá. El siguiente paso, en el archivo de configuración del proyecto, se introduce los parámetros de la base de datos a la que se va a conectar, es necesario introducir el motor que se va a utilizar, en este caso se trata de djongo y el nombre de la base de datos que se ha creado, *analizadorTematicoDB*.

Una vez realizada la conexión con la base de datos que se va a utilizar en el proyecto, se ha de realizar una migración con los cambios realizados usando la sentencia *python manage.py migrate*. En la base de datos se pueden observar las migraciones del proyecto. Se crea el super usuario, que será el utilizado durante toda la herramienta, tanto en la parte de administración, como su uso normal. Ya conectada la base de datos y creado el super usuario, es el momento de crear los modelos de la base de datos en el proyecto Django.

6.4. Diseño Web

En este apartado se encuentra la explicación del desarrollo realizado para el diseño web de la herramienta.

6.4.1. Front-End

El desarrollo del Front-End permite la interacción con la herramienta de forma digital utilizando HTML, CSS y JavaScript.

El Lenguaje de Marcas de Hipertexto, o como sus siglas HTML, es utilizado para el desarrollo del contenido de las páginas que se desarrollan para la aplicación. En estas se indican los diferentes contenidos que se han creado con la utilización de etiquetas para la identificación de los diferentes elementos. Las Hojas de Estilo en Cascada o CSS, se utilizan para el diseño de las páginas que se han generado con HTML. Es utilizado para los colores de la aplicación, el estilo de letra, la posición de los elementos que se encuentran. Para el diseño también se ha utilizado una biblioteca multiplataforma conocida como Bootstrap en su versión 4. Esto permite técnicas sencillas para que la herramienta sea *responsive* o adaptables a diferentes tamaño de páginas.

Para poder hacer uso de Bootstrap, es necesaria la instalación de la biblioteca en el proyecto, para ello, se instala Django-bootstrap4 versión 1.1.1. Una vez instalada esta versión, es necesario indicar a la aplicación que se ha instalado y que se va a utilizar.

En el archivo de configuración del proyecto, indicado anteriormente como *settings*, se observan las aplicaciones instaladas, en este momento es necesario indicar que se ha instalado la aplicación de bootstrap añadiendo en las aplicaciones la sentencia *bootstrap4*. El último paso es introducir la sentencia *CRISPY_TEMPLATE_PACK = 'bootstrap4'* en el archivo de configuración. Una vez indicado todo esto, se puede utilizar JavaScript y jQuery de bootstrap.

Otro aspecto a destacar usado en esta sección, es la librería Chart.js. Es una librería JavaScript de código abierto que permite la creación de gráficas para la visualización de los datos. Esta librería es utilizada en la sección de visualización de los datos recolectados donde se observan las diferentes gráficas que se han generado. Las funciones generadas están en la parte de los guiones o script que se han desarrollado para cada vista.

Para el desarrollo de la imagen generada con las palabras más utilizadas encontradas en los tuits recopilados, se ha utilizado el proyecto *wordcloud* en su versión 1.7.0. Para generar la nube de palabras, el proyecto utiliza la biblioteca *nltk* en su versión 3.5. Es la encargada de recoger en el texto todas las palabras que aparecen y realizar un conteo de apariciones. Genera una imagen con las palabras y dependiendo de su número de apariciones se ilustra con un tamaño u otro.

6.4.2. Back-End

El backend es la parte del desarrollo de la herramienta que se encarga de toda la lógica para que funcione correctamente. Para la explicación se van a desarrollar los siguientes puntos:

- Views

Cada una de las páginas que se han desarrollado en el frontend, están incluidas en el archivo *views* y se definen a través de funciones o en clases de Python dependiendo de su uso. Todas las vistas están definidas para que su uso solo se permita una vez que el usuario está con sesión iniciada. Para ello, en las funciones se utiliza un decorador *@login_required*, y en las clases se hereda de la clase *LoginRequiredMixin*. Django permite el uso de las clases cuando se está utilizando un modelo de la aplicación. Para hacer uso de las clases definidas en las vistas de Django, es necesario importarlas desde las vistas genéricas encontrándose en el proyecto *django.views.generic*. A través de esta, se pueden importar *ListView*, para poder visualizar una lista de modelos, *DetailView* para observar los detalles de cada objeto del modelo. *UpdateView* permite la actualización de cada objeto del modelo; *CreateView*, obtener una vista para la creación de un objeto del tipo modelo que se le indique. Se ha creado una clase que hereda de *ListView* para observar la lista de temas.

```
#Ver lista de Temas
class TemaListView(LoginRequiredMixin, ListView):
    model = Tema
```

Código 6. Clase que hereda de ListView.

En el Código 6 se observa cómo se ha implementado la clase para la visualización de los temas. Aparece el atributo *model* que es el utilizado para indicar el modelo al que pertenece. Para hacer una referencia sobre un archivo HTML para poder usar la lista de temas es necesario que el archivo tenga el nombre *tema_list.html*. Si se desea cambiar el nombre y referenciar a otro archivo diferente, se utiliza el atributo *template_name* seguido del nombre del archivo. De la misma forma que se ha creado la clase heredada de *ListView*, se realiza para la clase heredada de *DetailView*.

La vista para actualizar un tema hereda de *UpdateView*. Al igual que en la clase anterior es necesario indicar el modelo sobre el que se está trabajando. Se incluye el formulario que se va a utilizar con el atributo *form_class*. En el Código 7 se observa esta clase. También se observa una función utilizada para cargar una página una vez enviado el formulario si ha sido satisfactorio.

```

721 class TemaUpdateView(LoginRequiredMixin, UpdateView):
722     model = Tema
723     form_class = TemaUpdateForm
724     template_name = 'herramienta/tema_update_form.html'
725     #Hay que editar el success url para obtener la clave del objeto
726     def get_success_url(self):
727         return reverse_lazy('tema-detalle-view', kwargs={'pk':self.object.id})
728 
```

Código 7. Clase que hereda de UpdateView.

Las funciones para las vistas, se definen como una función de Python. Es necesario obtener como parámetro la solicitud web que viene dada por *request* y devuelve una respuesta web. Dentro de una función se puede introducir sentencias Python. Si se desea obtener un parámetro en el contenido HTML se define un *contexto*. Este contexto es un diccionario de Python, tiene una clave y un valor. El valor es el parámetro que se encuentra en la función y la clave es el parámetro que se utiliza dentro del contenido HTML. La función devuelve una respuesta web. Véase el Código 8 que trata sobre una vista para observar las tareas, en este caso devuelve el parámetro *request*, la vista a la que pertenece la función, es decir el HTML y el contexto.

```

54 @login_required
55 def tareas_view(request):
56
57     tareas = Task.objects.all()
58     tareasCompletas = CompletedTask.objects.all()
59     fechaHoy = datetime.date.today()
60
61     context = {
62         'tareas':tareas,
63         'completas':tareasCompletas,
64         'fechaHoy':fechaHoy
65     }
66
67     return render(request, 'herramienta/tareas.html', context)

```

Código 8. Vista sobre una función.

- URL

Para poder acceder a estas vistas en el navegador, es necesario la definición de URL. Django busca entre todos los patrones definidos en los archivos *urls.py*, y usa la primera que coincida. Cada URL llevará asociada una función para procesar la petición. El patrón de cada URL viene definida mediante expresiones regulares. Se destacan, el inicio con el símbolo ^, el fin indicado con \$. Si hay que señalar un tipo de dígito numérico a través de \d, entre otras. Otro aspecto a destacar es la necesidad de capturar parámetros. Por

ejemplo, si se necesita capturar una clave del tema, que es un parámetro numérico, se captura con la expresión regular `(?P<pk>\d+)`, indicando que se busca dígitos numéricos y que en la función este parámetro se representa mediante `pk`. El siguiente parámetro que se le indica a la URL, es la función a la que pertenece. Se observa un ejemplo de URL en el Código 9, en la cual, con la expresión regular, observamos el detalle del tema al que pertenezca esa clave primaria.

```
url(r'^analizador/tema/(?P<pk>\d+)$', TemaDetailView.as_view(), name='tema-  
detalle-view'),
```

Código 9. Ejemplo URL.

- Queryset

En cada una de las vistas de funciones en las que es necesario conocer el tema y la categoría, para poder visualizarla o hacer uso de la misma, se ha realizado una función denominada `datos(pk)`. Es necesario pasar como parámetro la clave primaria del tema al que pertenece. Dentro de esta función se hace uso de las *queryset*. Una *queryset* es una colección de objetos de la base de datos, se obtienen a partir de un *model manager*, que es una interfaz preparada para hacer consultas. Por defecto, todos los modelos tienen el *manager objects*. Para poder acceder a los datos de la base de datos, es necesario importar el modelo y obtenerla mediante la función `object.all()`, que contiene un objeto *queryset* con todos los datos del modelo. Permite realizar diferentes operaciones de búsqueda con su filtro, crear otro objeto, ordenarlos y contarlos, entre otros. Una vez encontrado el tema necesario, la función devuelve la clave, el nombre del tema y la *queryset* de las categorías que posee.

- Estadísticas

La herramienta ofrece la visualización de las estadísticas realizadas sobre los datos obtenidos. Se diferencian diferentes vistas; una vista en la cual, se pueden observar todas las categorías que tiene el tema y otra para cada categoría, también diferenciando una vez dentro de las diferentes vistas si pertenece a búsqueda *actual*, *histórica* o *actual vs histórica*. Sobre la vista individual de cada categoría, se muestra una primera gráfica que trata de un tipo *queso* o *pie* con el porcentaje de polaridad de los tuits de esa categoría. Se realiza una búsqueda de los tuits en la base de datos, obteniendo así el número total que se tiene. Una vez hecho esto, se diferencia entre las tres polaridades que contempla la herramienta y se realiza un porcentaje sobre su aparición. La siguiente gráfica es una evolución de la opinión mediante los días. Se recogen todos los tuits de la categoría, se observan los que pertenecen al mismo día, se realiza una media aritmética para conocer la media de la opinión perteneciente a ese día y se coloca en un diccionario, el cual, la clave es la fecha y el valor pertenece a la media calculada. El último aspecto que aparece en esta visualización es la nube de palabras. Se recoge el texto de todos los tuits y se calcula las palabras más repetidas, excluyendo espacios, preposiciones, artículos y posesivos que no son relevantes para esta imagen. Una vez creada, se introduce en el contexto con una URL para poder acceder a ella. La vista sobre las estadísticas en la que se pueden observar todas las categorías, utiliza la misma idea que se ha tomado para la anterior. Aparece una primera gráfica sobre el tema con el porcentaje de polaridad, continúa con una gráfica de barras, aparecen todas las categorías indicando el nivel del porcentaje de polaridad de cada una por separado, para

poder observar, cuál de ellas es la mejor valorada. El último aspecto es la evolución de la opinión por días apareciendo todas las categorías. Esto es útil, porque se puede conocer cuando hubo un cambio de polaridad por días, ya que, si alguna categoría pertenece a una empresa, se puede conocer si algún cambio ha sido bien recibido por la comunidad de la red social.

- Tareas en segundo plano

Una de las utilidades de la herramienta es la recopilación de información que se publica en la red social. Para ello, se han creado diferentes vistas para poder realizar una configuración de las búsquedas. Para poder realizar las búsquedas, se ha utilizado un proyecto existente en Django, el cual se denomina *django-background-tasks* en su versión 1.2.5. En la vista para la configuración de la búsqueda, se añade una sentencia, la cual, es una llamada a una función que posee un decorador para la utilización de las tareas en segundo plano. Esta tarea es la encargada de recopilar los tuits sobre la configuración de búsqueda enviada. Este decorador está definido como *@Background()*. En las definidas en la herramienta aparece un parámetro que se utiliza para programar la tarea para que empiece a los 5 segundos una vez invocada, este parámetro se define como *schedule=5*. Esto crea un objeto en la base de datos llamado *tasks*, contiene la tarea e información relevante a ella, como la fecha en la que se ha creado, la programación que tiene, si se ha de repetir, si tiene algún fallo, entre otros. Para poder ejecutar las tareas en segundo plano, es necesario abrir una consola que esté en la dirección del proyecto y ejecutar la sentencia *python manage.py process_tasks*. Una vez ejecutada la tarea, esta se elimina de la base de datos y se crea otro objeto llamado *completed task* en la base de datos con la información de la tarea que se acaba de ejecutar.

Test y pruebas

7.1. Test

Django permite la realización de pruebas unitarias que son útiles para la comprobación del perfecto funcionamiento de lo realizado en la herramienta. Para poder hacer uso de las pruebas unitarias, es necesario utilizar el módulo de la biblioteca estándar de Python llamada *unittest*, accediendo a ella a través de la sentencia *from django.test import TestCase*. Es necesario crear una clase que herede de la clase importada *TestCase*; en el caso de las pruebas realizadas sobre la herramienta se llamará *herramienta_test_class*. El primer paso para realizar las pruebas es implementar una función de establecer el inicio con los datos creados. Cuando se ejecuta una prueba, esta crea una base de datos por defecto con los objetos que se han creado en el inicio y cuando acabe de realizar las pruebas borra esta base de datos. A continuación se enumeran las pruebas realizadas:

- Tema y categoría.

En la siguiente prueba se crearán dos categorías y un tema que tenga esas dos categorías.

```
@classmethod
def setUpTestData(cls):
    categoria = Categoria.objects.create(nombre="Jazztel",
    palabras_clave="jazztel")
    categoria1 = Categoria.objects.create(nombre="Movistar",
    palabras_clave="movistar")
    Tema.objects.create(nombre="ADSL", palabras_clave="adsl",
    categorias=[categoria, categoria1])
```

Código 10. SetUp 1.

La prueba se encargará de recuperar la categoría uno, con el nombre de *Jazztel* y se comprobará que esa categoría pertenece al tema *ADSL*. Obsérvese que se genera un conjunto con el nombre de las categorías del tema y se comprueba que el nombre de la categoría este en ese conjunto.

```
def test_categoria_in_tema(self):
    nombre_categoria = Categoria.objects.get(pk=1).nombre
    categorias_tema = Tema.objects.get(pk=1).categorias
    categorias_nombre = []
    for categoria in categorias_tema:
        categorias_nombre += [categoria.nombre]
    self.assertTrue(nombre_categoria in categorias_nombre)
```

Código 11. Test categoría en tema.

Se ejecuta la prueba y se observa que da correcto, con lo que, se ha creado correctamente las categorías, el tema y que pertenece a ese tema.

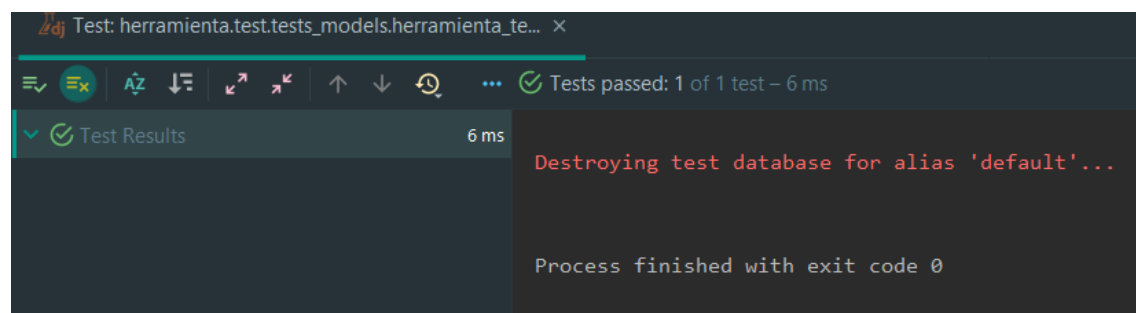


Imagen 31. Test correcto.

- Tema con Tweets.

En la siguiente prueba se crean dos tweets pertenecientes al tema ADSL y se comprobará qué se encuentra. Para ello, la prueba recogerá el nombre del tema ADSL a través de su clave única y realizará un filtrado de tweets buscando por nombre del tema. El resultado obtenido de tweets han de ser dos, con lo que la prueba se obtiene satisfactoriamente.

```
def test_tweets_in_tema(self):
    nombre_tema = Tema.objects.get(pk=1).nombre
    tweets = Tweet.objects.filter(tema=nombre_tema)
    cont = tweets.count()
    self.assertEqual(cont, 2)
```

Código 12. Test tweets en tema.

- Tema sin Tweets.

Esta prueba es parecida a la anterior, en este caso se genera un tema que no posee ningún tweet, con lo que al realizar la prueba de búsqueda no se obtiene ninguno, con lo que, la cantidad es cero. La prueba se obtiene satisfactoriamente.

```
def test_tweets_notin_tema(self):
    nombre_tema = Tema.objects.get(pk=2).nombre
    tweets = Tweet.objects.filter(tema=nombre_tema)
    cont = tweets.count()
    self.assertEqual(cont, 0)
```

Código 13. Test tema sin tweets.

7.2. Pruebas

En esta sección se realizan pruebas desde la propia herramienta. Se realizarán pruebas de creación de temas, categorías en esos temas, búsquedas de tuits históricas con el motor de análisis de TextBlob y MeaningCloud y tuits actuales. También se comprobará diferentes vistas.

- Tema y categoría.

Para crear un nuevo tema, es necesario dirigirse a la sección de *Analizador* y una vez ahí, seleccionar crear nuevo tema. Aparecerá un formulario con los atributos necesarios para crear el tema, una vez introducidos, guardaremos el tema.

Imagen 32. Prueba nuevo tema.

Esto dirigirá al usuario hacia la página inicial del analizador; se observa que se ha creado satisfactoriamente; se crean varios temas más para observar cómo se ve una lista de temas.

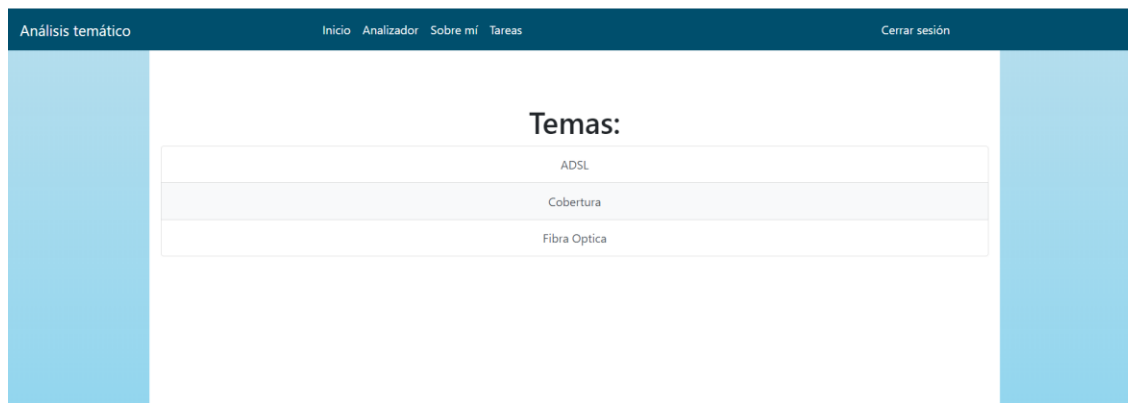


Imagen 33. Prueba vista temas.

Si se desean añadir más categorías a los temas, se entra en la página del tema en la sección de editar tema, que dejará crear una categoría por edición.

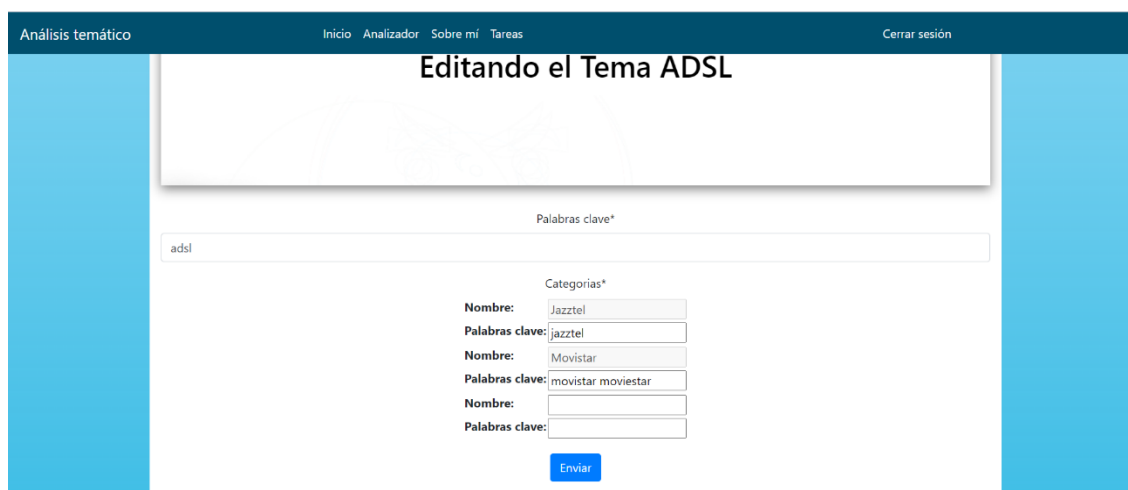


Imagen 34. Editar tema, añadir categoría.

- Búsqueda histórica TextBlob.

Ya se han realizado la creación del tema y de las categorías, el siguiente punto es la búsqueda histórica con el analizador TextBlob, que dirige hacia la configuración de búsqueda a través de la categoría que se desee.

Imagen 35. Búsqueda histórica TextBlob.

Una vez configurada la búsqueda, la herramienta generará una tarea en segundo plano para realizar la búsqueda.

```
Entre 2020-06-01 y 2020-06-05 se han encontrado 8 de 10
Termina el proceso
█
```

Imagen 36. Tarea segundo plano 1.

Como se observa en el terminal (Imagen 26), el proceso ha finalizado correctamente. Se observa que se han encontrado 8 tuits entre esas fechas indicadas. Si se observan los tuits pertenecientes a la búsqueda realizada, se ve que de esos 8 tuits encontrados, 4 de ellos pertenecen exactamente al tema y categoría fijados, debido a que la API de tweepy devuelve tuits que contengan las palabras del tema o de la categoría o de ambas y la herramienta se encarga de recoger tan solo los que contengan las palabras del tema y de la categoría.

Imagen 37. Observar tuits TextBlob.

- Búsqueda histórica MeaningCloud.

De la misma manera que se ha realizado la búsqueda anterior, es necesario realizar una configuración de búsqueda. En este caso en el formulario, es necesario seleccionar que el motor de análisis es MeaningCloud. Como la fecha de la recolección es la misma que la fecha anteriormente, los tuits recopilados son los mismos, lo único que cambia es el análisis de sentimientos realizado, debido a que se realiza con otro motor.



Imagen 38. Observar tuits MeaningCloud.

- Búsqueda actual.

Para realizar la búsqueda actual, es necesario introducir en el formulario la fecha hasta la que se desea recopilar los tuits. Cuando llega a esa fecha, la herramienta termina la recopilación. Cabe destacar que el análisis de sentimientos se realiza con TextBlob como se ha indicado durante todo el documento.

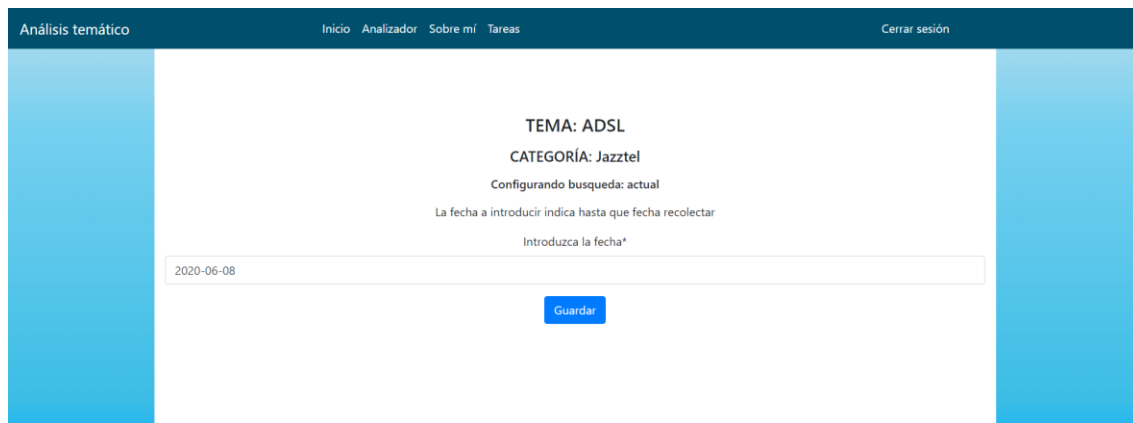
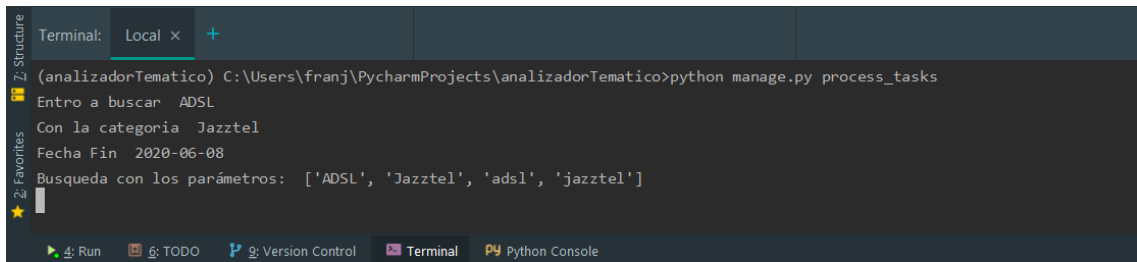


Imagen 39. Búsqueda actual

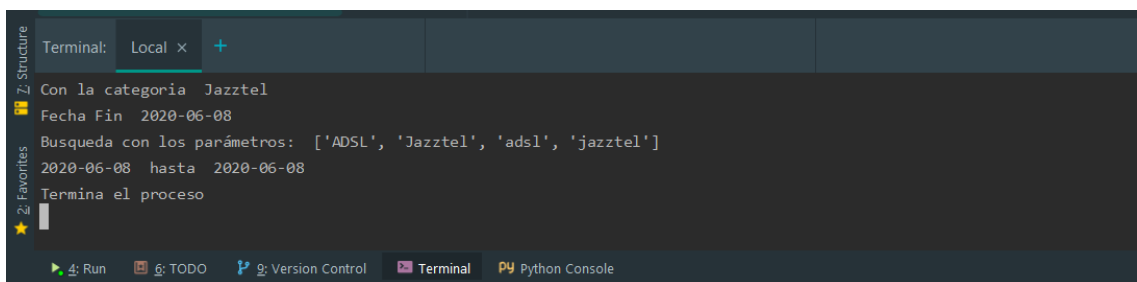
Se guarda la búsqueda actual y esto genera una tarea en segundo plano. En la Imagen 40 se muestra cómo se visualiza en el terminal.



```
Terminal: Local x +
(analizadorTematico) C:\Users\franj\PycharmProjects\analizadorTematico>python manage.py process_tasks
Entro a buscar ADSL
Con la categoria Jazztel
Fecha Fin 2020-06-08
Busqueda con los parámetros: ['ADSL', 'Jazztel', 'adsl', 'jazztel']
```

Imagen 40. Terminal inicio búsqueda actual.

Cuando llega la fecha fin, aparece que termina el proceso en segundo plano. Tal y como se observa en el terminal que está ejecutándose, aparece de nuevo la búsqueda que se está realizando debido a que cada vez que recolecta un tuit vuelven a aparecer los términos de búsqueda utilizados.



```
Terminal: Local x +
Con la categoria Jazztel
Fecha Fin 2020-06-08
Busqueda con los parámetros: ['ADSL', 'Jazztel', 'adsl', 'jazztel']
2020-06-08 hasta 2020-06-08
Termina el proceso
```

Imagen 41. Terminal fin búsqueda actual.

Para finalizar esta prueba, se observan los tuits que se han recolectado durante esta búsqueda. Se observan los últimos dos tuits, uno negativo y otro positivo.



Imagen 42. Visualización tuits actuales.

8

Conclusiones

8.1. Problemas encontrados

El problema principal ha sido la limitación en el uso de las API externas que se han utilizado en el proyecto. La API de MeaningCloud solo permite 20.000 consultas de análisis y por ese motivo se ha tenido que introducir una mejora añadiendo como motor de análisis la librería TextBlob. También destacar que esta biblioteca solo permite el análisis en inglés, por lo que, para poder hacer uso de ella, es necesario la traducción del texto que se recoge del tweet y analizarlo. Esto puede seleccionar errores en la traducción. Otro aspecto a destacar, es el uso limitado de la API tweepy. Cuando se hace uso de ella aparecen errores durante la retransmisión debido a que sobrepasan los límites. En estos casos es necesario controlar esos errores producidos, pero estos problemas, son internos de Twitter relativos a los permisos cuando se hace a través de la API gratuita. También, a la hora de recoger los tuits que ya han sido publicados, se observa que en selecciones no se recogen el número de tuits que el usuario desea recolectar, debido a problemas de limitaciones que no se pueden solucionar por consulta.

8.2. Líneas futuras

En este apartado se describen las líneas futuras de este proyecto para su mejora.

- Introducción de un mapa:

A lo largo de la recopilación de los tuits, se observa que un campo de la base de datos hace referencia a la posición geográfica donde se publicó. Inicialmente se pensó realizar un mapa geográfico para representar dónde se ha realizado la recopilación y dependiendo del lugar, representarlo de un color en función de la polaridad dominante en la región. No se ha podido realizar, debido a que no todos los tuits poseen el campo de localización, puesto que pueden tener desactivada la localización del dispositivo desde donde se envió el tuit. Uno de los datos que ofrece la API de Twitter es

información relevante del usuario que publica el tuit como información relativa a su ciudad indicada en el perfil. La información de la ciudad es el nombre de dicha ciudad, la localidad y el país a la que pertenece. Para la creación de un mapa en la que se refleja la polaridad en cada zona, se tiene que poner en común los datos geográficos dados en coordenadas en el tuit y la información de la ciudad que se obtiene del perfil. Se tienen en cuenta ambas debido a que la creación del perfil de Twitter pudo realizarse en una ciudad, pero la opinión pertenece a la localización del tuit. Para poner en común estos parámetros se tiene que observar todas las zonas que se quieran representar en el mapa y relacionar las coordenadas geográficas con las ciudades o localidades dependiendo de cómo se desee realizar el mapa. Se referencia las coordenadas geográficas con las localidades, de ese modo, se obtiene un punto en común para poder desarrollar el mapa.

- Inicio de sesión de usuario:

Se ha implementado el inicio de sesión para poder utilizar la aplicación, pero solo se ha tenido en cuenta, que el uso de la aplicación se hará para un usuario tipo administrador. Una de las líneas futuras, es poder tener la opción de crear diferentes usuarios con diferentes permisos: Un administrador que posea todos los permisos sobre la administración de datos y usuarios, que ya ofrece Django; otro usuario que pueda realizar recolecciones y observar las estadísticas, pero no la administración; otro para solo recolección de datos; y un último perfil que solo pueda observar las estadísticas. Esto permite que la herramienta pueda ser utilizada en un entorno empresarial, donde los usuarios tienen diferentes perfiles técnicos.

- Despliegue en la web:

Relacionada con la opción anterior, esta herramienta también es útil para poder utilizarla en la web. Con la opción de crear usuario, esto permitiría que cada uno de los usuarios que se registren en la herramienta, tengan su propia vista de solo recopilación y observar las estadísticas de los datos que poseen para ellos y que la vista de administración sea solo una opción para el creador de la herramienta.

- Recopilación en otro idioma:

La herramienta desarrollada realiza la recopilación y el análisis en español. Sería interesante dejar que se pueda introducir en las búsquedas otra opción de idioma y que el análisis se realice en el idioma que pertenezca dicha búsqueda.

- Motor de análisis de sentimientos:

Una de las restricciones que se han observado es referente al análisis de sentimientos, debido a que la herramienta depende de la limitación que posee la API. Esta línea futura plantea la posibilidad de realizar un motor de análisis de sentimientos propio. A través del aprendizaje automático, se puede realizar un algoritmo que aprenda de diferentes análisis que se realizan de diferentes textos, introduciendo una selección de palabras positivas y negativas dependiendo de los temas introducidos. Esto permitiría predecir posibles comportamientos futuros. A través de diferentes bibliotecas de Python para el lenguaje natural, permitirá observar y dividir las palabras de los textos y así, introducirlos en el motor con sus polaridades correspondientes para ir entrenando y que en el momento que se realice la recopilación propia de los tuits, ya conozca las

diferentes polaridades que seleccionan en los textos. La estructura de polaridad mantendrá la vista en la herramienta con tres ramas, positiva, negativa y neutra. Con esta estructura se van observando comportamientos, y cuando ya se han recopilado una cantidad de datos importante, el algoritmo ofrecerá un tanto por ciento de posibilidad de predecir un comportamiento u otro.

8.3. Conclusiones

La herramienta resultante cumple con las especificaciones y requisitos que se contemplaron en el inicio del proyecto, tanto en el desarrollo de las vistas que se plantearon, como en la lógica interna del proyecto. Se hace una recopilación de los tuits que sean pertenecientes al tema y la categoría seleccionados, que es uno de los objetivos iniciales del proyecto, donde además las estadísticas que se obtienen realizan una labor de ayuda para la toma de decisiones. Estas permiten observar la opinión de los usuarios que ya han experimentado con los temas que el usuario desea conocer.

El lenguaje utilizado para el desarrollo de la herramienta ha sido Python. Durante el desarrollo se ha observado que Python tiene un potencial enorme, permite el desarrollo de páginas web, como la implementada. La innumerable cantidad de librerías que contiene, la versatilidad, su sencillez, y lo legible a nivel de código para los usuarios son otras de sus características. Como esta aplicación se basa en datos, el uso de Python ayuda a automatizar los procesos de analítica, extracción de datos y visualizarlos.

También se ha utilizado el framework Django. A través de su configuración, distribución de ficheros y conexión entre los mismos, se ha visto la sencillez que ofrece para la realización de un proyecto web, facilitando el desarrollo tanto del back end como del front end, tal y como, se ha realizado en la herramienta. Django permite la reutilización de las aplicaciones que se crean para otros proyectos, tiene un contenido administrativo visual y la conexión con la base de datos no tiene complejidad.

Otro aspecto a destacar ha sido la utilización de una base de datos no relacional como es MongoDB. Se observa la escalabilidad que posee, su tolerancia a fallos, facilidad de administración, gracias a la aplicación que se instala durante el inicio de instalación de MongoDB llamada MongoDB Compass Community.

Referencias

- [1] L. L. y. S. V. Bo Pang, Thumbs up? Sentiment Classification using Machine Learning, Ithaca, NY 14853 USA: <http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf>, 2002.
- [2] J. M. W. a. R. F. B. a. T. P. O'Harat, Development and Use of a Gold-Standard Data Set for, Asheville, NC 28804-8511 : <https://www.aclweb.org/anthology/P99-1032.pdf>, 1990.
- [3] A. Berasategi, Sarcasm detection with NLP, Web: <https://towardsdatascience.com/sarcasm-detection-with-nlp-cbff1723f69a>, 2015.
- [4] Talkwalker, Página Talkwalker, Web: https://www.talkwalker.com/es/analisis-de-redes-sociales?gclid=Cj0KCQjw9ZzzBRCKARIsANwXaeKgQPlpdZJxKxeHty7oBFuD-glWY5cOLmbRp_tJyAKsYir2NrsfBZMaAlX-EALw_wcB, 2020.
- [5] Brand24, Página Brand24, Web: <https://brand24.com/>, 2020.
- [6] Keyhole, Página Keyhole, Web: <https://keyhole.co/>, 2020.
- [7] Rapidminer, Página Rapidminer, Web: <https://rapidminer.com/>, 2020.
- [8] E. Change, Extreme Programming Explained, Boston, EEUU: Addison-Wesley Professional, 2000.
- [9] N. y. Takeuchi, The New New Product Development Game, Web: <https://hbr.org/1986/01/the-new-new-product-development-game>, 1986.
- [10] d. l. H. L. Alvarez, Metodos ágiles y scrum, España: Anaya, 2012.
- [11] Softeng, Proceso y roles de Scrum, Web: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>, 2020.
- [12] Geekytheory, Página Geekytheory, Web: <https://geekytheory.com/que-es-una-api-rest-y-para-que-se-utiliza>, 2020.
- [13] B. open, Página BBVAOpen4u, Web: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>, 2020.
- [14] R. Fielding, Architectural Styles and the I of Network-based Software Architectures, CALIFORNIA, IRVINE: DOCTOR de Filosofía en Computación, 2000.
- [15] MeaningCloud, Página MeaningCloud, Web: <https://www.meaningcloud.com/es/sentiment-analysis>, 2020.
- [16] MonkeyLearn, Página MonkeyLearn, Web: <https://monkeylearn.com/text-classifiers/>, 2020.
- [17] I. W. N. L. Understanding, Página IBM, Web: <https://cloud.ibm.com/docs/services/natural-language-understanding/index.html>, 2020.

- [18] Indico, Página Indico, Web: <https://indico.io/blog/docs/indico-api/text-analysis/sentiment-analysis/>, 2020.
- [19] AYLIEN, Página AYLIEN, Web: <https://aylien.com/> , <https://docs.aylien.com/textapi/#language-support>, 2020.
- [20] T. PreCeive, Página TheySay, Página TheySay: <https://apidemo.theysay.io/>, 2020.
- [21] G. C. N. Language, Página Google Cloud, Web: <https://cloud.google.com/natural-language?hl=es-419#section-2>, 2020.
- [22] M. A. T. Analytics, Página Microsoft Azure, Web: <https://azure.microsoft.com/es-es/services/cognitive-services/text-analytics/>, 2020.
- [23] TextBlob, Página TextBlob, Web: <https://pypi.org/project/textblob/> , <https://textblob.readthedocs.io/en/dev/>, 2020.
- [24] VADER, Página VADER, Web: <https://pypi.org/project/vaderSentiment/>, 2020.
- [25] R. B. a. L. H. Alec Go, Sentiment140, Web: <http://help.sentiment140.com/api>, 2020.
- [26] P. Rivera, Tweepy, Web: <http://docs.tweepy.org/en/latest/>, 2020.
- [27] Python, Página Python, Web: <https://www.python.org/>, 2020.
- [28] Django, Página Django, Web: <https://www.djangoproject.com/>, 2020.
- [29] MongoDB, Página MongoDB, Web: <https://www.mongodb.com/>, 2020.
- [30] PyCharm, Página PyCharm, Web: <https://www.jetbrains.com/es-es/pycharm/>, 2020.
- [31] JetBrains, Página JetBrains, Web: <https://www.jetbrains.com/>, 2020.

Apéndice A

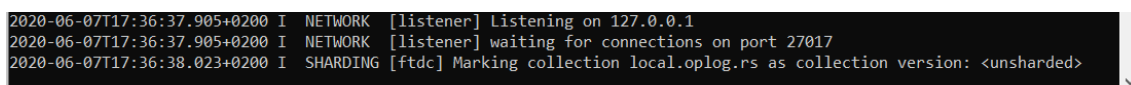
Manual de Instalación

Introducción

A lo largo de este apéndice se explicará el proceso de instalación de la herramienta. El ejemplo de instalación se hará sobre el sistema operativo Windows 10. Se comenzará con el desarrollo de puesta en marcha de MongoDB. A continuación, se plantea el uso de PyCharm, el entorno de desarrollo que se ha utilizado durante este proyecto. Una vez realizada toda la instalación de estos, es necesario la introducción del código de la aplicación que se encuentra en la plataforma GitHub. Repositorio <https://github.com/FranGamezRuiz/analizadorTematico>.

Instalación de MongoDB.

Para la instalación de MongoDB, el primer paso es descargar desde su página oficial el instalador <https://www.mongodb.com/download-center/community>. Se selecciona la versión 4.2.5, que es la que se ha utilizado, o la versión actual del momento para el sistema operativo de la máquina desde donde se esté realizando la instalación y el paquete, en este caso MSI. Una vez descargado, ejecutamos el instalador, y aparecerá una guía a seguir para el proceso. Se selecciona el botón *Next* y aparece una nueva ventana para seleccionar la instalación que se desea, en este caso, *complete*. Aparecerá entonces la opción de instalar *MongoDB Compass Community* funcional para un entorno visual, y comenzará a instalarse. Una vez instalado, se tendrá que ejecutar el comando *mongod*, para iniciar el servidor. Es necesario ir al directorio donde se haya instalado, que suele ser C:\Program Files\MongoDB\Server\4.2\bin. Una vez copiada la ruta, se añade a símbolos del sistema dentro de las variables de entorno del usuario. Se selecciona *Path* y se copia la ruta donde se instaló MongoDB. Al aceptar los cambios ya permitirá la ejecución de los comandos anteriores, pero aún no se puede iniciar. Dentro de la ruta del directorio local es necesario la creación de una carpeta llamada *data*; dentro de esta carpeta se crea otra llamada *db*, y dentro de ella se guardarán las bases de datos que se creen para MongoDB. Si se ejecuta el servidor de MongoDB, se observa el puerto por el que se puede acceder a él, que en este caso, la dirección es 127.0.0.1:27017.



```
2020-06-07T17:36:37.905+0200 I NETWORK [listener] Listening on 127.0.0.1
2020-06-07T17:36:37.905+0200 I NETWORK [listener] waiting for connections on port 27017
2020-06-07T17:36:38.023+0200 I SHARDING [ftdc] Marking collection local.oplog.rs as collection version: <unsharded>
```

Imagen A 1. Dirección MongoDB.

Desde otro terminal, se puede ejecutar el comando *mongo*, para iniciar una terminal de MongoDB para crear, editar, borrar bases de datos. En la instalación, también se realizó la instalación de *MongoDB Compass Community*, una interfaz para usar las bases de datos. Se ejecuta ambos comandos en dos terminales diferentes y ejecuta la aplicación, se observa el menú inicial. El requisito para conectarse al servidor de MongoDB, es conocer la dirección y el puerto, que una vez introducidos permiten conectarse a él y hacer uso de su interfaz.

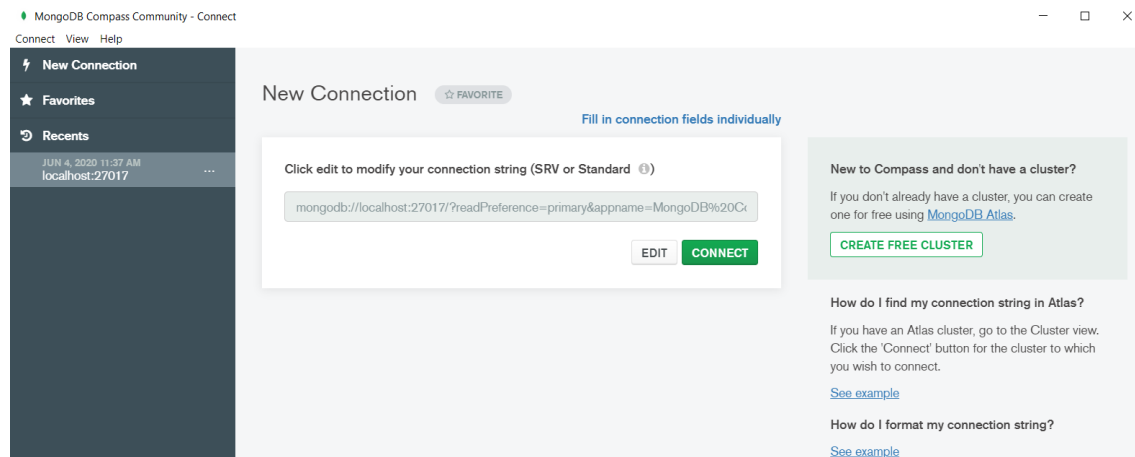


Imagen A 2. Inicio Mongo Compass.

Se usa el botón *create database* para crear la base de datos que se va a utilizar en la herramienta, denominada en este caso, *analizadorTematicoDB*.

Instalación Django.

Para utilizar la aplicación con el framework de Django, es necesario la instalación de Python. A través de su página oficial <https://www.python.org/downloads/> se obtiene la versión 3.8.1. Si se desea crear un entorno virtual para el proyecto, se recomienda descargar la versión de Anaconda Python 3.7 que se puede encontrar en su página web <https://www.anaconda.com/products/individual>. Una vez instalado Python, se puede realizar la instalación por dos métodos: usando el entorno virtual utilizado en el desarrollo, PyCharm con Anaconda (<https://www.jetbrains.com/pycharm/promo/anaconda/>) o simplemente desde el terminal. Para poder usar la versión PyCharm profesional, es necesaria una clave, esta puede ser conseguida gracias a las licencias que otorga JetBrains a través del enlace <https://account.jetbrains.com/licenses/assets>.

Ejecución de la herramienta.

El siguiente paso a seguir es realizar la instalación de los requisitos del proyecto. Para ello, solo se ha de ejecutar la sentencia *pip install -r requirements.txt*, con esto, se instalará todo lo necesario para que el proyecto pueda ser utilizado. Se abre un terminal para ejecutar el comando de MongoDB *mongod*. En otro terminal, ejecutamos la sentencia *python manage.py runserver* y se ejecuta la herramienta, para poder observarla desde el navegador. La ejecución muestra la dirección, tan solo ha de introducirse en un explorador.

Apéndice B

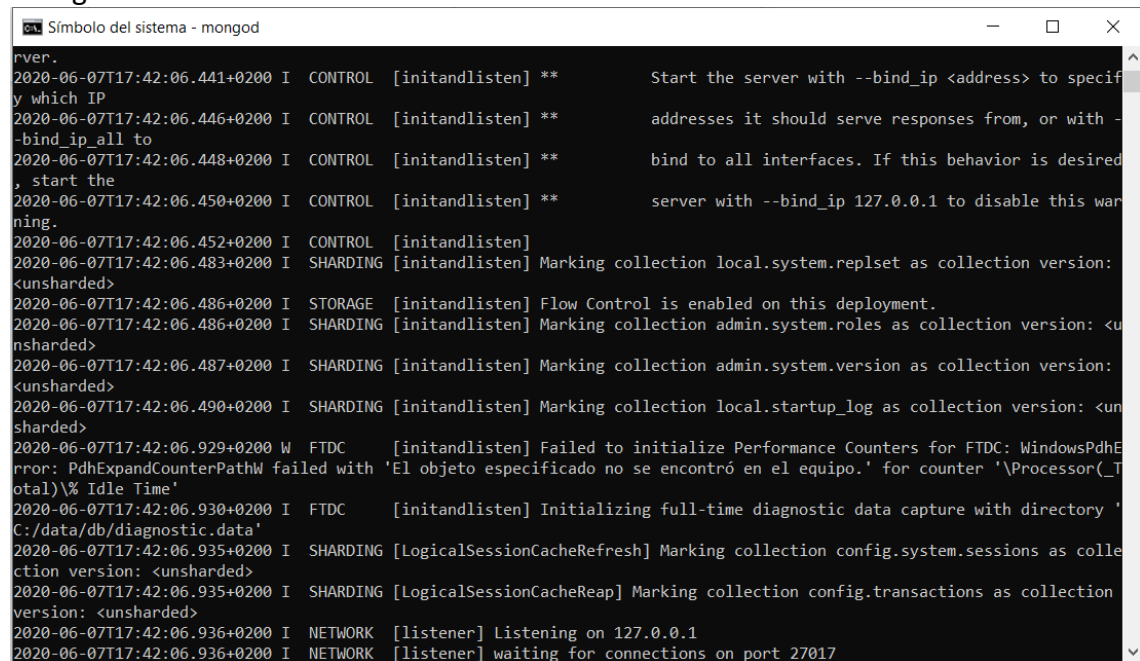
Guía de uso

Introducción

En este apéndice se describe una guía para que cualquier usuario pueda dar uso a la herramienta que se ha desarrollado en este Trabajo de Fin de Grado. Se explica cómo preparar el entorno para ejecutar el servidor, así como las vistas para el uso de la herramienta y una breve explicación sobre dónde se encuentran las diferentes utilidades.

Preparación del entorno

Se abre un terminal en el que se ejecuta la sentencia *mongod*, para iniciar el servidor de MongoDB.



```
Símbolo del sistema - mongod
2020-06-07T17:42:06.441+0200 I CONTROL [initandlisten] ** Start the server with --bind_ip <address> to specify which IP
2020-06-07T17:42:06.446+0200 I CONTROL [initandlisten] ** addresses it should serve responses from, or with --bind_ip_all to
2020-06-07T17:42:06.448+0200 I CONTROL [initandlisten] ** bind to all interfaces. If this behavior is desired, start the
2020-06-07T17:42:06.450+0200 I CONTROL [initandlisten] ** server with --bind_ip 127.0.0.1 to disable this warning.
2020-06-07T17:42:06.452+0200 I CONTROL [initandlisten]
2020-06-07T17:42:06.483+0200 I SHARDING [initandlisten] Marking collection local.system.replset as collection version: <unsharded>
2020-06-07T17:42:06.486+0200 I STORAGE [initandlisten] Flow Control is enabled on this deployment.
2020-06-07T17:42:06.486+0200 I SHARDING [initandlisten] Marking collection admin.system.roles as collection version: <unsharded>
2020-06-07T17:42:06.487+0200 I SHARDING [initandlisten] Marking collection admin.system.version as collection version: <unsharded>
2020-06-07T17:42:06.490+0200 I SHARDING [initandlisten] Marking collection local.startup_log as collection version: <unsharded>
2020-06-07T17:42:06.929+0200 W FTDC [initandlisten] Failed to initialize Performance Counters for FTDC: WindowsPdhError: PdhExpandCounterPathW failed with 'El objeto especificado no se encontró en el equipo.' for counter '\\Processor(\\Total)\\% Idle Time'
2020-06-07T17:42:06.930+0200 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'C:/data/db/diagnostic.data'
2020-06-07T17:42:06.935+0200 I SHARDING [LogicalSessionCacheRefresh] Marking collection config.system.sessions as collection version: <unsharded>
2020-06-07T17:42:06.935+0200 I SHARDING [LogicalSessionCacheReap] Marking collection config.transactions as collection version: <unsharded>
2020-06-07T17:42:06.936+0200 I NETWORK [listener] Listening on 127.0.0.1
2020-06-07T17:42:06.936+0200 I NETWORK [listener] waiting for connections on port 27017
```

Imagen A 3. Ejecución mongod.

Una vez ejecutado el servidor, se puede observar que está esperando conexiones en el puerto 27017. Eso quiere decir, que ya se ha ejecutado correctamente. Se procede a abrir el proyecto en PyCharm, donde se ha realizado el desarrollo de la herramienta. En la parte superior derecha, aparece un botón para ejecutar el programa, aquí se puede realizar una configuración de la ejecución. Véase en la Imagen A 4 la configuración que se ha tomado durante el desarrollo de este apéndice. En este se observa el intérprete que se ha utilizado, Python 3.7, con el entorno virtual creado llamado analizadorTematico.

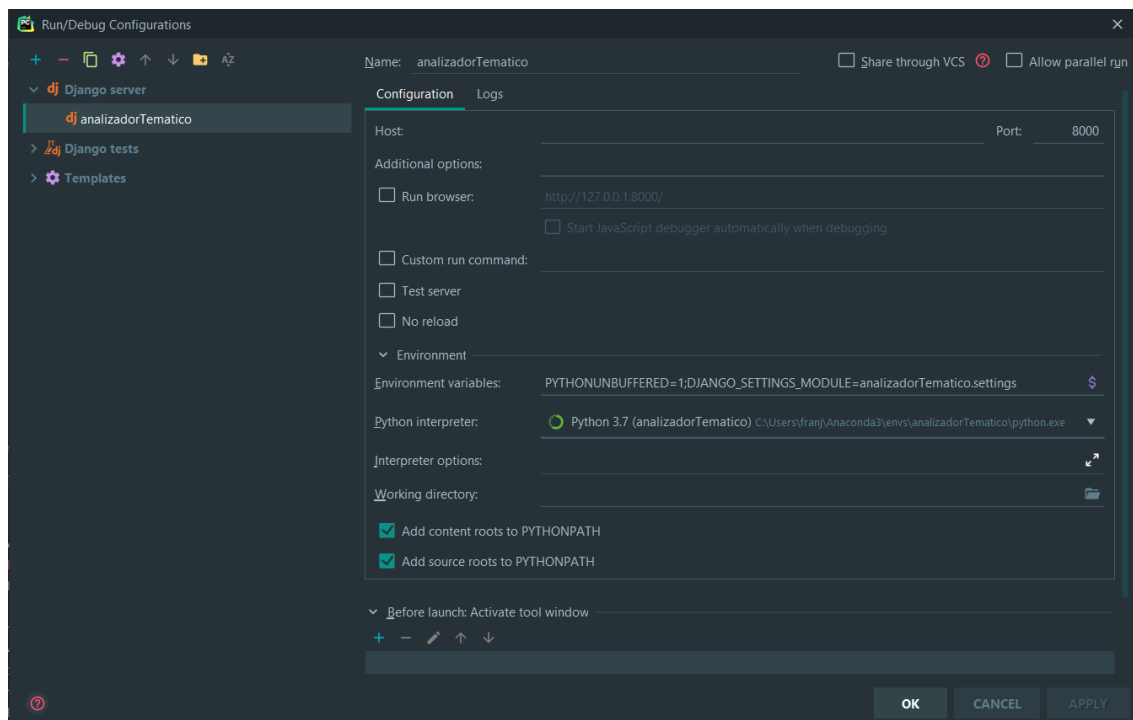


Imagen A 4. Configuración PyCharm.

A través del botón, se ejecuta el proyecto correctamente. En el caso de que no se haya realizado la instalación con PyCharm, desde la carpeta donde se haya clonado el proyecto, se abre un terminal y se ejecuta la sentencia *python manage.py runserver*. Esto hace que aparezca la dirección para acceder a la herramienta.

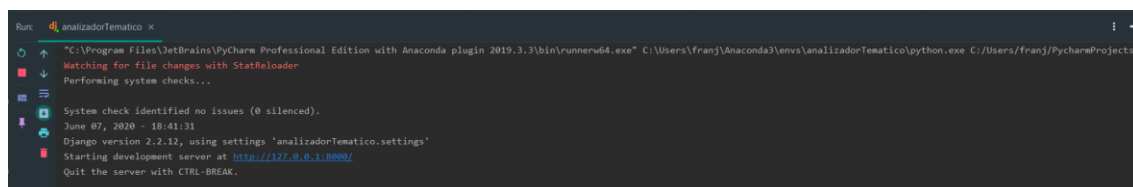
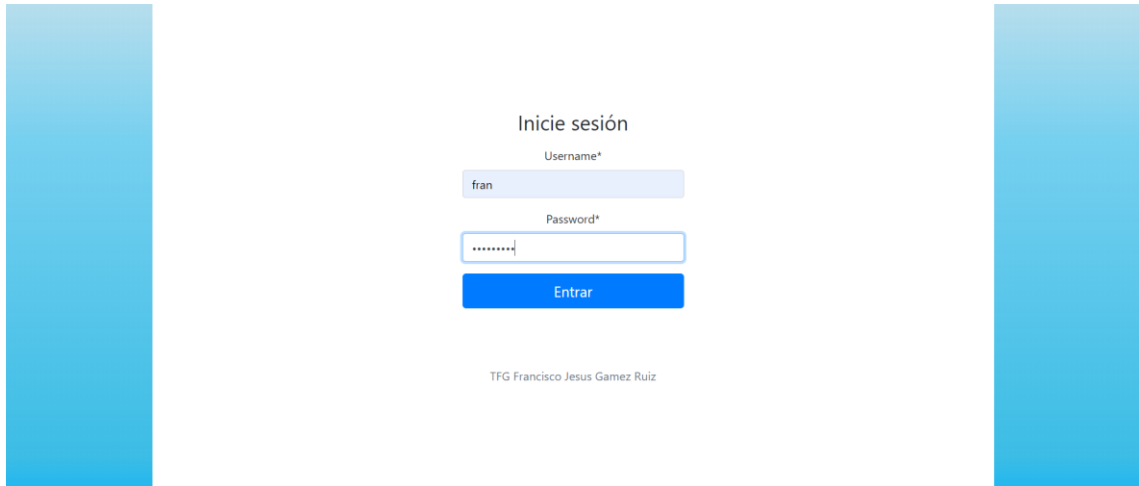


Imagen A 5. Ejecución del proyecto.

En otro terminal en la misma dirección del proyecto o desde el terminal que ofrece el entorno, se ha de ejecutar la sentencia *python manage.py process_tasks*, para la ejecución de las tareas en segundo plano referentes a la recolección. Una vez hecho esto, ya se puede acceder sin problema a la dirección que dice el terminal, en este caso, <http://127.0.0.1:8000>. También se puede acceder con `localhost:8000`, debido a que se ejecuta en local.

Herramienta.

La primera vista que ofrece la herramienta es el inicio de sesión. Para poder iniciar, se deben de introducir las credenciales de usuario *fran* y contraseña *franAdmin*.



The image shows a login form titled "Inicie sesión" centered on a white background with blue vertical bars on the sides. The form contains two input fields: "Username*" with the value "fran" and "Password*" with masked characters "*****". Below these fields is a blue button labeled "Entrar". At the bottom of the form, the text "TFG Francisco Jesus Gamez Ruiz" is displayed.

Imagen A 6. Inicio de sesión.

Aparece la página inicial, en la cual, se puede encontrar un logo de la herramienta junto a la explicación sobre sus funcionalidades. Si se sigue avanzando hacia abajo, se puede observar una explicación sobre análisis de sentimientos y la justificación de usar Twitter.



Imagen A 7. Inicio de la herramienta.



Imagen A 8. Análisis de sentimientos.

Como se puede observar en la Imagen A 8, aparece un barra de navegación en la parte superior. En ella se indican las diferentes páginas que hay en la herramienta. *Analizador* hace referencia a la página para realizar la recolección y análisis, *sobre mí* explica quién ha realizado la herramienta y *tareas*, son las tareas en segundo plano de recolección. También aparece la opción de *cerrar sesión*.



Imagen A 9. Sobre mí.

El siguiente paso a observar es, en el apartado de *Analizador*, los diferentes temas que hay en la herramienta, una explicación sobre qué es un tema, qué es el objeto tweet, los parámetros que lo componen y un botón para poder exportar los tweets que hay en la base de datos en formato CSV.

Para empezar a utilizar la herramienta, es necesario la creación del tema. Realizando un desplazamiento hacia la parte inferior, se observa un botón para la creación del tema, se accede a él y se crea el tema que se desee realizar la búsqueda. Se observa en la Imagen A 10.



Imagen A 10. Explicación Tema.

En este ejemplo, se va a crear un tema llamado *Fibra óptica* para buscar opiniones que los usuarios de Twitter tengan sobre ella. Para buscar opiniones tan solo de fibra óptica, se utiliza una categoría con el nombre de *general*, que es la palabra reservada para ello. Esto buscará en la red social las publicaciones que contengan la palabra "fibra óptica" y las almacenará en la base de datos. El siguiente paso es conocer otras categorías, se generan dos, *Jazztel* y *Movistar*, debido a que se desea conocer las opiniones de estas categorías, con el objetivo, por ejemplo, de un cambio de contratación en la vivienda y se quiere conocer la opinión para seleccionar la mejor.

Imagen A 11. Creando Tema.

Una vez creado el tema con una categoría, el paso siguiente es editar el tema para poder añadir de una en una las categorías que se desee. En el menú de analizador, se encuentra una entrada sobre el tema *Fibra óptica*, justo en este lugar es donde se podrán encontrar todos los temas que se creen.

En la entrada del tema se pueden observar varias opciones: primero se encuentran las entradas a las categorías que se han creado para el tema, junto a ello, se puede observar que hay un botón para la edición del tema. Esto permite la creación de una nueva categoría o la edición de las palabras clave de búsqueda del tema y de las categorías existentes. En este ejemplo, queda claro que las categorías sirven para conocer

diferentes opiniones sobre el tema teniendo en cuenta dos compañías de telecomunicaciones. La última parte es la opción para la visualización de las estadísticas del tema; aparecerá un botón para visualizarlo. Más adelante se observará este, debido a que no se han encontrado datos en este momento.



Imagen A 12. Categorías.

Si se quiere recopilar información para conocer la opinión de los usuarios, tan solo sobre *fibra óptica*, se selecciona la opción de la categoría *general*. Una vez abierta esa vista, se podrán ver varios apartados (Imagen A 13). En el primero de ellos, es para la configuración de las búsquedas, que permite la opción de *histórica* o *actual*. El segundo y último, da una explicación al usuario de que se trata esas búsquedas para que el usuario pueda elegir la mejor de ella.



Imagen A 13. Búsquedas.



Imagen A 14. Explicación de las búsquedas.

Se va a realizar un ejemplo sobre la búsqueda de datos históricos. Se configura la búsqueda histórica, se selecciona y aparece un formulario. En él el usuario puede seleccionar el número de tuits que desea recopilar, y dos fechas, una de inicio y otra de fin. La de inicio representa la fecha más antigua desde la que se desea recopilar, esta no puede superar los siete días debido al límite de la API de Twitter; y la de fin es la más cercana al día actual. Para realizar la búsqueda debe haber mínimo un día de diferencia entre ambas. Por último, se selecciona el motor de análisis que se desea utilizar. En este caso, se va a utilizar *TextBlob* ya que no tiene limitaciones, la otra opción es *MeaningCloud*, que sí posee limitaciones, 20.000 peticiones al mes. Por esa razón, en el formulario se recomienda usarla cuando la recolección no supere los 10.000. Esto se puede ver en la Imagen A 15.

Imagen A 15. Configuración de búsqueda.

Una vez guardada la configuración, la herramienta generará una tarea en segundo plano. Si se observa el terminal en el que se ejecutó la sentencia `python manage.py process_tasks`, se visualiza que ha empezado a generar información sobre la tarea de recopilación que se ha guardado. Si se desea observar las tareas que hay en la herramienta se puede ir a la opción de *tareas*.

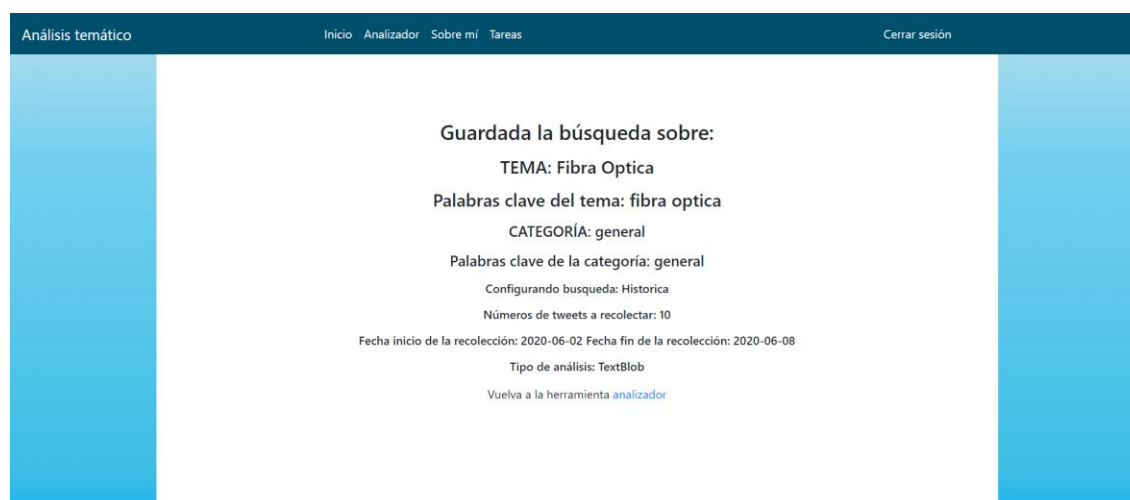


Imagen A 16. Guarda la búsqueda configurada.

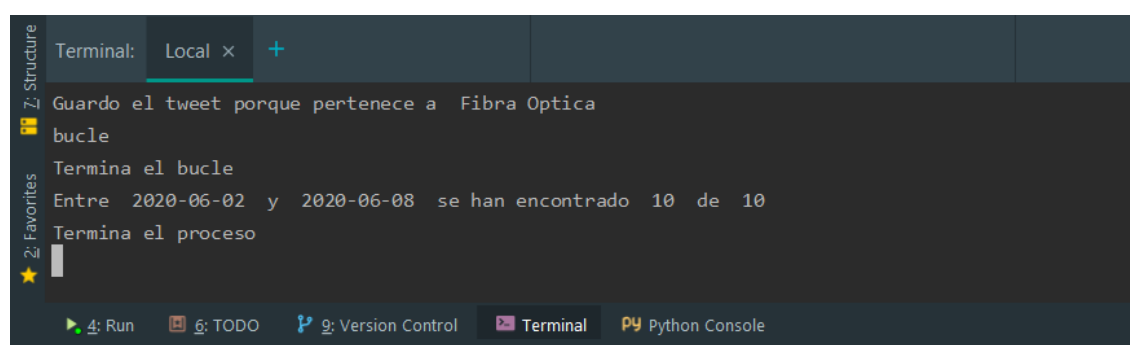


Imagen A 17. Se observa que se ha completado la tarea.

Para la recopilación de tuits sobre las categorías creadas anteriormente, *Jazztel* y *Movistar*, se realiza de la misma forma como se ha realizado para la categoría *general*. El siguiente paso es observar las estadísticas que se han generado gracias a la recolección. Para ello, hay que irse hacia la sección de *Dashboard*.

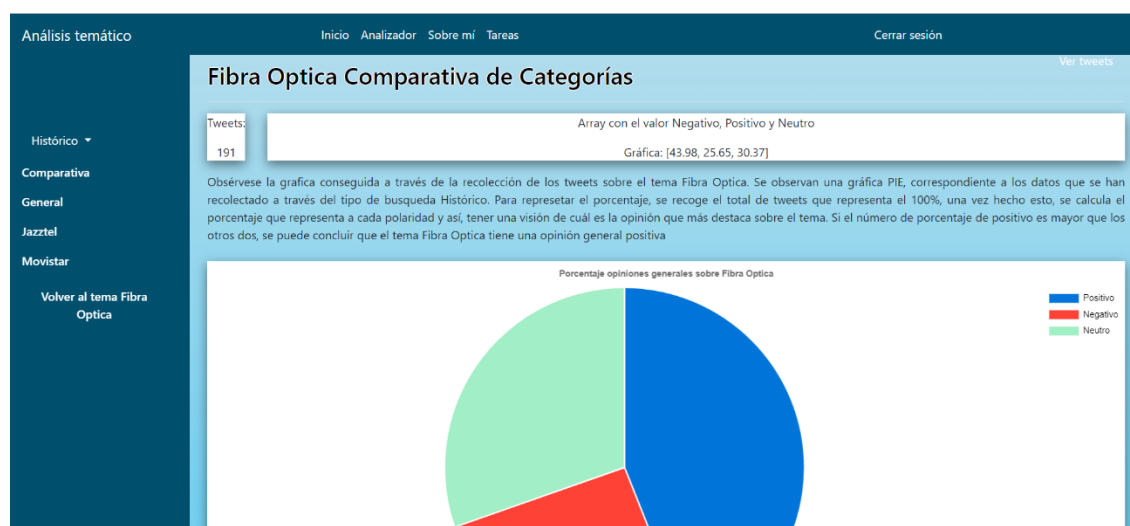


Imagen A 18. Vista Dashboard.

En la Imagen A 18 se puede observar un menú en la parte de la izquierda para una navegación fluida entre las búsquedas y categorías. La primera opción permite la selección de las búsquedas: *histórica*, *actual* o *actual vs histórica*. Las siguientes opciones son sobre las categorías. *Comparativa*, en la que aparece la página de

comparación de todas las categorías que hay en el tema que se está visualizando. *General*, donde se visualiza la opinión del tema en general, tanto con los tuits que se han recolectado con la forma de la categoría *general*, como en cualquier búsqueda. Esto es debido a que aunque pertenezca a algún tema, esa opinión también se ha de tener en cuenta para una opinión general. Y por último, se puede seleccionar cada una de las categorías que se han creado.

Visualícese la primera vista que aparece como *Fibra Óptica Comparativa de Categorías*. Aparecen en la parte superior dos datos, uno referente al número de tuits que se han recopilado para esta sección y el segundo referente a una lista visual del número del porcentaje de polaridades de la información recopilada. Con estos porcentajes, se ha generado la siguiente gráfica. Esta representa el porcentaje de las polaridades sobre el tema en general, que viene heredada de la sección *general*. Permite una primera visualización de las opiniones que poseen los usuarios de la red social. Con este dato, el usuario puede conocer cuando se publican opiniones sobre el tema, qué características tienen. Por ejemplo, en este caso, cuando se hacen referencias sobre el tema *Fibra óptica*, los usuarios son positivos en un 43.98%, al ser el más alto de los tres, se puede destacar, que *Fibra óptica* es algo positivo entre los usuarios.

La siguiente gráfica que aparece en esta sección, es referente a las polaridades de todas las categorías que pertenecen al tema. Esto permite una mayor visualización sobre cuál de ellas ha obtenido mayor porcentaje en una polaridad que al usuario necesite conocer. En este caso, si se desea conocer cuál es el porcentaje positivo que más habitual en las categorías, se observa que *Jazztel*, tiene un mayor porcentaje frente a *Movistar*, con lo que podemos decir, que la opinión que se tiene de la *Fibra óptica* de *Jazztel* es generalmente positiva en mayor medida que la *Fibra óptica* de *Movistar*.



Imagen A 19. Polaridad por categoría.

La última vista que pertenece a esta opción, es una gráfica de líneas. Cada línea representa una categoría del tema, y en la gráfica se introducen las fechas en las que se ha recopilado los tuits. Para cada fecha, se calcula una media de la opinión en ella para que el usuario pueda conocer en qué fechas se ha cambiado la opinión de los usuarios. Esto sirve para conocer en qué momento se aumentó o disminuyó la positividad o negatividad de las opiniones y así poder inferir a qué se debe. Se puede ver con mayor perspectiva si afecta directamente las opiniones entre las categorías que hay en el tema.

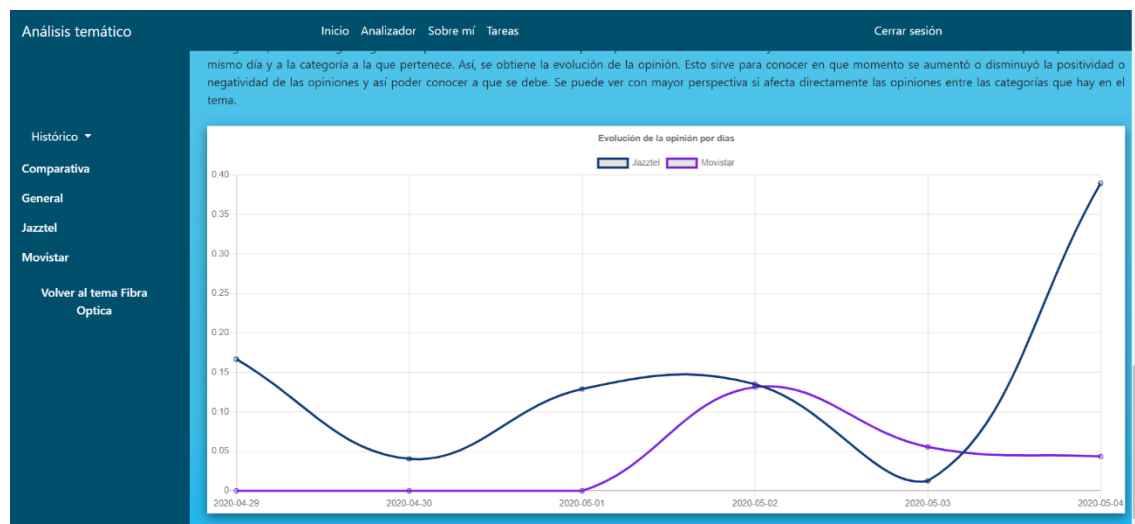


Imagen A 20. Evolución días por categoría.

Si se va al principio de la vista, se observa la opción de *Ver tweets*. Esto permite al usuario observar todos los tuits que pertenecen a la vista en la que se encuentra. En este caso, se observarán todos los tuits que pertenezcan al tema *Fibra óptica*. Junto a ello, aparece un botón para exportar los tuits en formato CSV.

ID	Tweet	Tema	Categoría	Tipo de Búsqueda	Tipo de Análisis	Polaridad	Fecha
ID:1257341342667214849	@movistar_es hace más de 3 semanas que estoy esperando un rutero para solucionar el problema... excepté recibir mensaje de bots no tengo nada. Dime @movistar_es te puedes moverte el culo o me voy por @jazztel_es porque pagar 130 euros la fibra que no funciona es una vergüenza!	Fibra Optica	Jazztel	historica	TextBlob	0.62500	2020-05-04
ID:1257311599208984577	@JosAnto86491506 ¡Hola José Antonio! Mi nombre es Eloy y te atiendo desde casa. Mencionarte, que la intención de Jazztel, siempre es la de ampliar y mejorar el despliegue de cobertura de fibra directa, para llegar al mayor número posible de clientes, que junto con los 1/4	Fibra Optica	Jazztel	historica	TextBlob	0.20000	2020-05-04
ID:1257307323220946944	Las novedades de Jazztel Fibra y Móvil https://t.co/5XAiQrGoW9 Gracias a @Ethic_ @MediayMarketing #mkdirecto #estilodevida	Fibra Optica	Jazztel	historica	TextBlob		

Imagen A 21. Visualizar tweets.

El siguiente paso es ver por categoría las vistas que se han generado en el dashboard. Se explicará solo la opción *general*, debido a que todas las categorías tienen la misma vista. Se observa que aparece el número de tuits que se han recopilado en esta sección y junto a esto, aparece el tipo de análisis que se ha utilizado y su cantidad correspondiente. En este caso, se han realizado búsquedas, tan solo con, el análisis *TextBolb*.

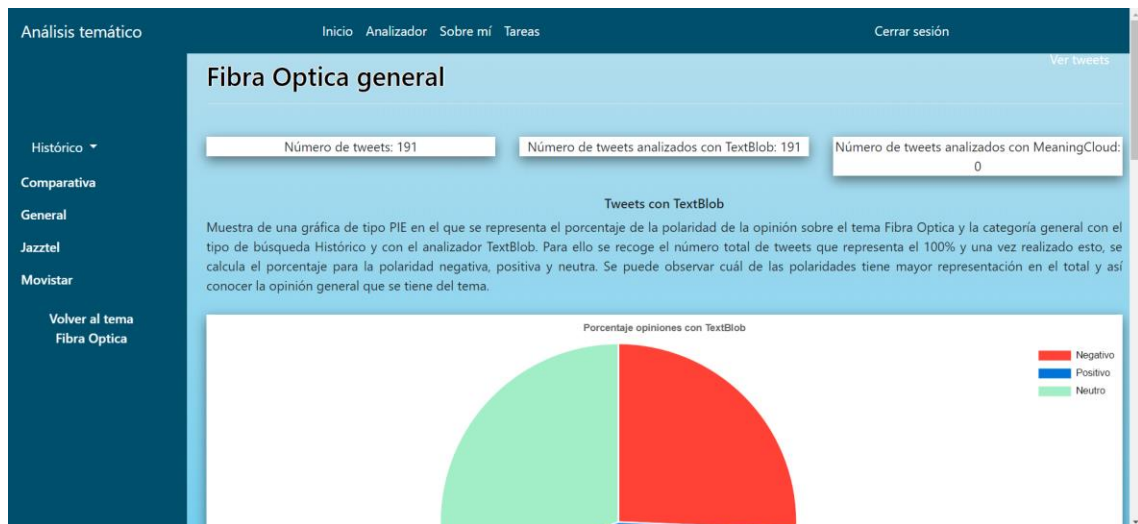


Imagen A 22. Vista general.

En la Imagen A 22 aparece una gráfica referente al porcentaje de polaridades, tal y como se ha visto en la sección anterior, con lo que, simplemente se muestra gráficamente. También se observa la gráfica de la evolución por día de la categoría en la que se encuentra, en este caso, se trata de la evolución de la opinión sobre *fibra óptica* en *general*.



Imagen A 23. Evolución opinión en general.

En la Imagen A 23 se observa que con el tiempo, la opinión que se recoge en *general* sobre *fibra óptica* ha caído aunque se ha mantenido positiva. Esto puede ser debido a que en el periodo de búsquedas ha tenido lugar la pandemia del COVID-19, por lo que, los usuarios han tenido problemas en conexiones de *fibra óptica* y han tomado represalias al opinar sobre este tema. El prohibir el cambio de compañía, ha enfurecido a muchos usuarios. También se han limitado recursos para evitar el colapso de las conexiones en el país. Gracias a la evolución de la opinión se ha podido conocer la razón de este cambio de opinión. También se ha generado una nube de palabras que a permitido conocer los términos más destacados cuando se habla sobre *fibra óptica* en la red social.



Imagen A 24. Nube de palabras.

Como se observa en la Imagen A 24, cuando se habla de *fibra óptica*, existen palabras positivas y negativas que se repiten, como por ejemplo *tienen*, *velocidad*, *novedades*. Como en la sección anterior, también se pueden visualizar tuits pertenecientes a *general*, donde aparecen los mismos tuits que antes, todos los recolectados sobre el tema. Se mostrará los tuits de la categoría *Movistar*, donde se puede observar que aparecen tanto los positivos, negativos como los neutros, diferenciados por iconos para facilitar la visualización.



Imagen A 25. Tuits movistar.



UNIVERSIDAD
DE MÁLAGA

| **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga